



**Escuela Superior
de Ingeniería**

ESCUELA SUPERIOR DE INGENIERÍA

GRADO EN INGENIERÍA INFORMÁTICA

**TUNICAPP: TU APLICACIÓN WEB PARA LA
GESTIÓN DE TU HERMANDAD**

AUTOR: JESÚS MANUEL GARCÍA LUZ

Cádiz, Junio 2020



**Escuela Superior
de Ingeniería**

ESCUELA SUPERIOR DE INGENIERÍA

GRADO EN INGENIERÍA INFORMÁTICA

**TUNICAPP: TU APLICACIÓN WEB PARA LA
GESTIÓN DE TU HERMANDAD**

DIRECTOR: CARLOS RIOJA DEL RÍO
AUTOR: JESÚS MANUEL GARCÍA LUZ

Cádiz, Junio 2020

Índice general

Índice	8
Índice de figuras	15
Índice de tablas	19
Índice de códigos	21
1. Introducción	23
1.1. Contexto	23
1.1.1. Motivación	23
1.1.2. Objetivo	23
1.1.3. Alcance	24
1.1.4. Enfoque metodológico	25
1.1.5. Normativas	25
1.1.6. Estructura de la memoria	25
1.1.7. Glosario	26
1.1.7.1. Acrónimos	26
2. Conceptos preliminares y estado del arte	27
2.1. Forma de trabajo actual	27
2.2. Conclusión	28
2.3. Investigación sobre herramientas	28
2.3.1. Java	28
2.3.2. JPA	29
2.3.3. JSF	29
2.3.4. PrimeFaces	29
2.3.5. Google Charts	30
2.4. Conclusión	30
3. Desarrollo del calendario	31
3.1. Fases	31
3.1.1. Fase primera: Inicio proyecto	31
3.1.2. Fase segunda: Trabajo de investigación	31
3.1.3. Fase tercera: Objetivos	31
3.1.4. Fase cuarta: Desarrollo	33
3.1.5. Fase quinta: Pruebas y depuración	33
3.1.6. Fase sexta: Conclusiones	34
3.2. Costes	34
3.2.1. Costes de recursos humanos	34
3.2.2. Costes de recursos materiales	34
3.3. Diagrama de Gantt	35
4. Descripción general del proyecto	37

4.1.	Descripción del proyecto	37
4.2.	Perspectiva del proyecto	37
4.3.	Metodología de desarrollo	38
4.4.	Módulos y Funcionalidades	40
4.4.1.	Inicio	40
4.4.2.	Inicio de Sesión	41
4.4.3.	Módulos del Administrador	42
4.4.3.1.	Personas	42
4.4.3.2.	Hermanos	45
4.4.3.3.	Control de Salida	48
4.4.3.4.	Estadísticas	52
4.4.3.5.	Túnicas	53
4.5.	Tipos de usuarios	56
4.6.	Interfaz de usuario	56
4.6.1.	Administrador	56
4.6.2.	Usuario	58
4.7.	Lenguajes de programación y tecnologías	59
4.7.1.	Java	59
4.7.2.	Ajax	59
4.7.3.	SQL	59
4.7.4.	Latex	59
4.7.5.	Selenium	60
4.8.	Herramientas	61
4.8.1.	JSF	61
4.8.1.1.	Porqué utilizar JSF	61
4.8.2.	JPA	61
4.8.3.	Primefaces	61
4.8.4.	Maven	62
4.8.5.	Tomcat Apache	62
4.8.6.	GitHub	62
5.	Análisis	65
5.1.	Requisitos	65
5.1.1.	Requisitos funcionales	65
5.1.2.	Requisitos de información	67
5.1.3.	Requisitos no funcionales	68
5.2.	Análisis del sistema	69
5.2.1.	Casos de uso	69
5.2.2.	Casos de uso - Entrar	70
5.2.3.	Casos de uso - Iniciar Sesión	70
5.2.4.	Casos de uso - Personas	71
5.2.5.	Casos de uso - Control de Salida	71
5.2.6.	Casos de uso - Hermanos	71
5.2.7.	Casos de uso - Túnicas	72
5.2.8.	Casos de uso - Reserva tu Control de Salida	72
5.2.9.	Casos de uso - Tus cuotas	73
5.2.10.	Casos de uso - Listado de Personas	74
5.2.11.	Casos de uso - Exportar a PDF el Listado de Personas	74
5.2.12.	Casos de uso - Exportar a Excel el Listado de Personas	75
5.2.13.	Casos de uso - Añadir Persona	75
5.2.14.	Casos de uso - Editar Persona	76
5.2.15.	Casos de uso - Eliminar Persona	78
5.2.16.	Casos de uso - Listado de Túnicas	79
5.2.17.	Casos de uso - Exportar a PDF el Listado de Túnicas	80
5.2.18.	Casos de uso - Exportar a Excel el Listado de Túnicas	80
5.2.19.	Casos de uso - Añadir Túnica	81

5.2.20. Casos de uso - Editar Túnica	82
5.2.21. Casos de uso - Eliminar Túnica	83
5.2.22. CU-033: Listado de Hermanos	85
5.2.23. Casos de uso - Exportar a PDF el Listado de Hermanos	85
5.2.24. Casos de uso - Exportar a Excel el Listado de Hermanos	86
5.2.25. Casos de uso - Añadir Hermano	86
5.2.26. Casos de uso - Editar Hermano	87
5.2.27. Casos de uso - Eliminar Hermano	89
5.2.28. CU-033: Listado de Control de Salida	90
5.2.29. Casos de uso - Exportar a PDF el Listado de Control de Salida	91
5.2.30. Casos de uso - Exportar a Excel el Listado de Control de Salida	91
5.2.31. Casos de uso - Añadir Control de Salida	92
5.2.32. Casos de uso - Editar Control de Salida	93
5.2.33. Casos de uso - Eliminar Control de Salida	94
5.3. Diagrama de Secuencias	96
5.3.1. Diagrama de Secuencia - Entrar	97
5.3.2. Diagrama de Secuencia - Iniciar Sesión	98
5.3.3. Diagrama de Secuencia - Personas	99
5.3.4. Diagrama de Secuencia - Control de Salida	100
5.3.5. Diagrama de Secuencia - Hermanos	101
5.3.6. Diagrama de Secuencia - Túnicas	102
5.3.7. Diagrama de Secuencia - Reserva tu Control de Salida	103
5.3.8. Diagrama de Secuencia - Tus cuotas	104
5.3.9. Diagrama de Secuencia - Listado de Personas	105
5.3.10. Diagrama de Secuencia - Exportar a PDF el Listado de Personas	106
5.3.11. Diagrama de Secuencia - Exportar a Excel el Listado de Personas	107
5.3.12. Diagrama de Secuencia - Añadir Persona	108
5.3.13. Diagrama de Secuencia - Editar Persona	109
5.3.14. Diagrama de Secuencia - Eliminar Persona	110
5.3.15. Diagrama de Secuencia - Listado de Túnicas	111
5.3.16. Diagrama de Secuencia - Exportar a PDF el Listado de Túnicas	112
5.3.17. Diagrama de Secuencia - Exportar a Excel el Listado de Túnicas	113
5.3.18. Diagrama de Secuencia - Añadir Túnica	114
5.3.19. Diagrama de Secuencia - Editar Túnica	115
5.3.20. Diagrama de Secuencia - Eliminar Túnica	116
5.3.21. Diagrama de Secuencia - Listado de Hermanos	117
5.3.22. Diagrama de Secuencia - Exportar a PDF el Listado de Hermanos	118
5.3.23. Diagrama de Secuencia - Exportar a Excel el Listado de Hermanos	119
5.3.24. Diagrama de Secuencia - Añadir Hermano	120
5.3.25. Diagrama de Secuencia - Editar Hermano	121
5.3.26. Diagrama de Secuencia - Eliminar Hermano	122
5.3.27. Diagrama de Secuencia - Listado de Control de Salida	123
5.3.28. Diagrama de Secuencia - Exportar a PDF el Listado de Control de Salida	124
5.3.29. Diagrama de Secuencia - Exportar a Excel el Listado de Control de Salida	125
5.3.30. Diagrama de Secuencia - Añadir Control de Salida	126
5.3.31. Diagrama de Secuencia - Editar Control de Salida	127
5.3.32. Diagrama de Secuencia - Eliminar Control de Salida	128
5.3.33. Diagrama de Secuencia - Estadísticas	129
5.4. Modelo de clases inicial	130
6. Diseño	131
6.1. Diseño de la herramienta	131
6.1.1. Base de Datos	131
6.1.2. Modelo de clases	132
6.1.3. Infraestructura	134
6.1.4. Vistas y formularios	134

6.1.4.1.	Vista General	134
6.1.4.2.	Vista principal del Hermano	135
6.1.4.3.	Vista principal del Administrador	135
6.1.4.4.	Alta de Control de Salida	136
7.	Implementación	139
7.1.	Programación en Java	139
7.1.1.	JPA	139
7.1.2.	JSF	141
7.1.3.	PrimeFaces	142
7.2.	Interfaz de la herramienta	142
7.3.	Funciones de interés sobre los componentes web	150
7.3.1.	Verificación de campos	151
7.4.	Métodos de interés	153
7.4.1.	Verificación de DNI y predicción de letra	153
7.4.2.	Encriptación y descriptación de la contraseña	153
7.4.3.	Envío de Email	155
7.4.3.1.	Documentación personalizada	163
7.4.4.	Pago mediante PayPal	166
7.5.	Accesos a la Base de Datos	171
8.	Pruebas y validación	175
8.1.	Pruebas unitarias	175
8.2.	Pruebas de integración	177
8.3.	Pruebas de aceptación	177
8.4.	Pruebas del sistema	177
8.4.1.	Pruebas Funcionales	177
8.4.2.	Pruebas No Funcionales	177
8.4.3.	Pruebas Automáticas	177
9.	Conclusiones y trabajo futuro	179
9.1.	Conclusiones sobre la herramienta	179
9.2.	Experiencia personal con el proyecto	181
9.3.	Trabajo futuro	181
9.3.1.	Mejoras	182
A.	Manual de instalación	183
A.1.	Java	183
A.2.	WAMP	184
A.3.	Eclipse	185
A.4.	Git	185
A.5.	Configuración de la herramienta	186
B.	Manual de desarrollador	189
B.1.	Estructura del código	189
B.1.1.	Java Resources	190
B.1.2.	Base	190
B.1.3.	Persistencia	191
B.1.4.	Negocio	195
B.1.5.	Helpers	197
B.1.6.	Controladores	197
B.1.7.	Test	197
B.1.7.1.	Unitarios	198
B.1.7.2.	Automatizados	198
C.	Manual de usuario	199
C.1.	Introducción	199

C.2. Inicio	199
C.3. Login	200
C.4. Pantalla principal Administrador	201
C.5. Personas	201
C.5.1. Listado de Personas	201
C.5.2. Añadir Persona	202
C.5.3. Editar Personas	203
C.5.4. Eliminar Persona	204
C.6. Túnicas	204
C.6.1. Listado de túnicas	204
C.6.2. Añadir Túnica	204
C.6.3. Editar Túnica	205
C.6.4. Eliminar túnica	206
C.7. Hermanos	206
C.7.1. Listado de Hermanos	206
C.7.2. Añadir Hermano	207
C.7.3. Editar Hermano	208
C.7.4. Eliminar Hermano	209
C.8. Control de Salida	209
C.8.1. Listado de Control de Salida	209
C.8.2. Añadir Control de Salida	209
C.8.2.1. Justificante Control de Salida	211
C.8.3. Editar Control de Salida	211
C.8.4. Eliminar Control de Salida	212
C.8.5. Estadísticas	213
C.9. Pantalla principal Usuario	214
C.9.1. Tus Cuotas	214
C.9.2. Reserva tu Control de Salida	214
Bibliografía	216

Índice de figuras

2.1. Java	28
2.2. JSF	29
2.3. PrimeFaces	30
2.4. Google Charts	30
3.1. Diagrama Gannt.	36
4.1. Modelo en cascada.	38
4.2. Bienvenid@ a tunicAPP.	40
4.3. Pantalla de login.	41
4.4. Listado de Personas.	42
4.5. Alta de Personas	43
4.6. Editar Persona.	44
4.7. Borrar Persona.	45
4.8. Listado de Hermanos.	46
4.9. Añadir Hermano	47
4.10. Borrar Hermano	48
4.11. Listado de Controles de Salida.	49
4.12. Añadir Control de Salida	50
4.13. Editar Control de Salida	51
4.14. Borrar Control de Salida	52
4.15. Estadísticas	53
4.16. Listado de Túnica.	54
4.17. Añadir Túnica	54
4.18. Editar Túnica	55
4.19. Borrar Túnica	55
4.20. Panel de Hermano	58
4.21. Logo Java	59
4.22. Logo Ajax.	59
4.23. Logo mySQL.	59
4.24. Logo Latex.	60
4.25. Logo Latex.	60
4.26. Logo JSF.	61
4.27. Logo JPA.	62
4.28. Logo PrimeFaces.	62
4.29. Logo Maven.	62
4.30. Logo TomCat Apache.	63
4.31. Logo GitHub.	63
5.1. Diagrama para Administrador.	68
5.2. Administrador: Diagrama de casos de uso.	69
5.3. Hermano: Diagrama de casos de uso.	69
5.4. DS01 - Entrar	97
5.5. DS02 - Iniciar Sesión	98

5.6.	DS03 - Personas	99
5.7.	DS04 - Control de Salida	100
5.8.	DS05 - Hermanos	101
5.9.	DS06 - Túnicas	102
5.10.	DS07 - Reserva tu Control de Salida	103
5.11.	DS09 - Tus Cuotas	104
5.12.	DS10 - Listado de Personas	105
5.13.	DS11 - Listado de Personas a PDF	106
5.14.	DS12 - Listado de Personas a Excel	107
5.15.	DS13 - Añadir Persona	108
5.16.	DS14 - Editar Persona	109
5.17.	DS15 - Eliminar Persona	110
5.18.	DS16 - Listado de Túnicas	111
5.19.	DS17 - Listado de Túnicas a PDF	112
5.20.	DS18 - Listado de Túnicas a Excel	113
5.21.	DS19 - Añadir Túnica	114
5.22.	DS20 - Editar Túnica	115
5.23.	DS21 - Eliminar Túnica	116
5.24.	DS22 - Listado de Hermanos	117
5.25.	DS23 - Listado de Hermanos a PDF	118
5.26.	DS24 - Listado de Hermanos a Excel	119
5.27.	DS25 - Añadir Hermano	120
5.28.	DS26 - Editar Hermano	121
5.29.	DS27 - Eliminar Hermano	122
5.30.	DS28 - Listado de Control de Salida	123
5.31.	DS29 - Listado de Control de Salida a PDF	124
5.32.	DS30 - Listado de Control de Salida a Excel	125
5.33.	DS31 - Añadir Control de Salida	126
5.34.	DS32 - Editar Control de Salida	127
5.35.	DS33 - Eliminar Control de Salida	128
5.36.	DS34 - Estadística	129
6.1.	Base de Datos local.	132
6.2.	Diagrama Final	133
6.3.	Inicio	134
6.4.	Panel del Hermano	135
6.5.	Panel del Administrador	135
6.6.	Alta de Control de Salida	136
6.7.	Validación de Alta Hermano	137
7.1.	Arquitectura Java	139
7.2.	Persistencia de JPA	140
7.3.	Diagrama básico de JSF	141
7.4.	Logo de PrimeFaces	142
7.5.	Estructura de la Interfaz	143
7.6.	Alta de Hermano.	150
A.1.	Bienvenido a Java	184
A.2.	Instalación de WAMP	184
A.3.	Nuestro IDE: Eclipse	185
A.4.	Un controlador de versiones... GIT	186
A.5.	Bash de GIT	186
A.6.	Importamos nuestro proyecto	187
A.7.	Añadimos Apache TomCat	187
B.1.	Estructura del proyecto.	189

B.2. Estructura Java del proyecto.	190
B.3. Arquitectura de paquetado Java.	190
B.4. Estructura Java: Base.	191
B.5. Estructura Java: Persistencia.	192
C.1. Pantalla de Inicio.	199
C.2. Pantalla de Login.	200
C.3. Pantalla de Inicio para rol Administrador.	201
C.4. Pantalla de Listado de Personas.	202
C.5. Pantalla para añadir Personas.	203
C.6. Pantalla de Listado de Túnicas.	204
C.7. Pantalla para añadir Túnicas.	205
C.8. Pantalla para editar la túnica.	206
C.9. Pantalla del listado de Hermanos.	207
C.10. Pantalla para añadir Hermanos.	208
C.11. Pantalla del Listado de Controles de Salida.	209
C.12. Pantalla para añadir un Control de Salida.	210
C.13. Justificante generado al Control de Salida.	211
C.14. Editar un Control de Salida	212
C.15. Estadística	213
C.16. Pantalla de Inicio para rol Usuario.	214
C.17. Pantalla de Tus Cuotas.	215

Índice de tablas

3.1. Costes de Recursos Humanos	34
3.2. Costes de Recursos Materiales	34
5.1. CU-001: Entrar	70
5.2. CU-002: Inicio de Sesión	70
5.3. CU-003: Personas	71
5.4. CU-004: Control de Salida	71
5.5. CU-005: Hermanos	71
5.6. CU-006: Túnicas	72
5.7. CU-007: Reserva tu Control de Salida	72
5.8. CU-008: Reserva tu Control de Salida	73
5.9. CU-009: Tus Cuotas	73
5.10. CU-010: Tus Cuotas	73
5.11. CU-011: Listado de Personas	74
5.12. CU-012: Exportar a PDF el Listado de Personas	74
5.13. CU-013: Exportar a Excel el Listado de Personas	75
5.14. CU-014: Añadir Persona	75
5.15. CU-015: Añadir Persona	76
5.16. CU-016: Editar Persona	76
5.17. CU-017: Editar Persona	77
5.18. CU-018: Editar Persona	77
5.19. CU-019: Eliminar Persona	78
5.20. CU-020: Eliminar Persona	78
5.21. CU-021: Eliminar Persona	79
5.22. CU-022: Listado de Túnicas	79
5.23. CU-023: Exportar a PDF el Listado de Túnicas	80
5.24. CU-024: Exportar a Excel el Listado de Túnicas	80
5.25. CU-025: Añadir Túnica	81
5.26. CU-026: Añadir Túnica	81
5.27. CU-027: Editar Túnica	82
5.28. CU-028: Editar Túnica	82
5.29. CU-029: Editar Túnica	83
5.30. CU-030: Eliminar Túnica	83
5.31. CU-031: Eliminar Túnica	84
5.32. CU-032: Eliminar Túnica	84
5.33. CU-033: Listado de Hermanos	85
5.34. CU-034: Exportar a PDF el Listado de Hermanos	85
5.35. CU-035: Exportar a Excel el Listado de Hermanos	86
5.36. CU-036: Añadir Hermano	86
5.37. CU-037: Añadir Hermano	87
5.38. CU-038: Editar Hermano	87
5.39. CU-038: Editar Hermano	88
5.40. CU-039: Editar Hermano	88

5.41. CU-040: Eliminar Hermano	89
5.42. CU-041: Eliminar Hermano	89
5.43. CU-042: Eliminar Hermano	90
5.44. CU-043: Listado de Control de Salida	90
5.45. CU-044: Exportar a PDF el Listado de Control de Salida	91
5.46. CU-045: Exportar a Excel el Listado de Control de Salida	91
5.47. CU-046: Añadir Control de Salida	92
5.48. CU-047: Añadir Control de Salida	92
5.49. CU-048: Editar Control de Salida	93
5.50. CU-049: Editar Control de Salida	93
5.51. CU-050: Editar Control de Salida	94
5.52. CU-051: Eliminar Control de Salida	94
5.53. CU-052: Eliminar Control de Salida	95
5.54. CU-053: Eliminar Control de Salida	95
5.55. CU-054: Estadística	96

Índice de códigos

7.1. Arquitectura Front del aplicativo	144
7.2. Componente Web de JSF	151
7.3. Growl de mensajes	151
7.4. EliminarHermano	152
7.5. validarNIEDNI()	153
7.6. encriptar()	153
7.7. desencriptar()	155
7.8. Clase MailBuilder	155
7.9. Clase MailClient	157
7.10. Clase BuilderPDF	163
7.11. Clase PayPalBuilder	167
7.12. Clase PayPalClient	168
7.13. XML API PayPal	170
7.14. Clase DataBaseBuilder	171
8.1. Clase AutoHermanos.java	175
B.1. Persistence.xml	191
B.2. Hermano.java	192
B.3. Pasos.java	195

Capítulo 1

Introducción

En este capítulo se abordarán el contexto, la motivación y los objetivos del proyecto referente al trabajo de fin de grado (TFG) realizado, así como otras características, como son su alcance, el enfoque metodológico utilizado y el resultado obtenido tras su realización. Se incluye también la estructura del presente documento, así como un glosario de términos.

1.1. Contexto

El presente trabajo de fin de grado ha sido desarrollado para cubrir una necesidad en la Hermandad del Perdón de Cádiz, en colaboración con su Hermano Mayor para la toma de requisitos. Se ha decidido utilizar tecnologías punteras en desarrollo demandadas en el actual mercado laboral para mejorar en dicho ámbito.

1.1.1. Motivación

Una de las razones principales del desarrollo fue la necesidad de implantar una herramienta que pudiese sustituir los actuales formularios a papel realizados mediante Word o formularios de Acrobat. Una hermandad, al fin y al cabo, es una congregación de personas y, como tal, es necesario gestionar, organizar y proteger la información de sus usuarios.

Dicha tarea de desarrollo, agilizará las labores de gestión en las salidas procesionales y altas de hermanos en las que, actualmente, tienen muchos problemas de organización. Se pretende realizar una aplicación eficiente, optimizada, agradable a la vista y sencilla haciendo hincapié en este último aspecto ya que los usuarios finales de la aplicación no son usuarios con mucha relación a la informática.

1.1.2. Objetivo

El proyecto de fin de grado TUNICAPP: TU APLICACIÓN WEB PARA LA GESTIÓN DE TU HERMANDAD tiene como objetivo principal la obtención de una herramienta para la gestión de la Hermandad, que agilice el proceso del tratamiento de los recursos humanos, eficacia en sus acciones y aporte soluciones para las dificultades obtenidas todos estos años en la gestión de la salida procesional. Además de cubrir la necesidad del cliente, el desarrollo y aprobado de este proyecto por parte del Tribunal de la Universidad de Cádiz permitiría la obtención del Título de Grado en Ingeniería Informática al autor Jesús Manuel García Luz, alumno de la Universidad de Cádiz y poder profundizar aún mas en las últimas tecnologías en desarrollo Java demandadas a nivel nacional.

1.1.3. Alcance

TUNICAPP: TU APLICACIÓN WEB PARA LA GESTIÓN DE TU HERMANDAD pretende ofrecer una mejora importante respecto a la forma de trabajo actual para la Junta de Gobierno de la Hermandad del Perdón, acortando así los tiempos en gestión, organización y asistencia para la formación de la Salida Procesional, aportando un tratamiento de la información efectivo, claro y conciso minimizando en la medida de lo posible el factor de error humano, mejorando así la calidad de trabajo de la Junta de Gobierno y facilitando a los ayudantes la labor organizativa del mismo.

Para ello, TUNICAPP: TU APLICACIÓN WEB PARA LA GESTIÓN DE TU HERMANDAD deberá cumplir el objetivo anteriormente citado, el cual se desglosa en las siguientes acciones:

- Rol de Administrador
 - Personas
 - Alta
 - Baja
 - Editar
 - Eliminar
 - Hermanos
 - Alta
 - Baja
 - Editar
 - Eliminar
 - Control de Salida
 - Alta
 - Baja
 - Editar
 - Eliminar
 - Estadísticas
 - Túnicas
 - Alta
 - Baja
 - Editar
 - Eliminar
- Rol de Usuario
 - Tus cuotas
 - Reserva tu Control de Salida

Además de estas funcionalidades, este proyecto ha de suministrar una sección para permitir el inicio de sesión del usuario y comprobar su rol.

1.1.4. Enfoque metodológico

Para este proyecto se utilizará una metodología modular, incremental y basada en proyectos debido al tipo de herramienta que se va a desarrollar, la cual requiere de un análisis de requisitos u objetivos por parte de la parte solicitante que han de ser cumplidos y cuyas exigencias y necesidades pueden ser modificadas durante el tiempo de desarrollo de la aplicación.

Además de esto, el posible incremento o cambio respecto a los objetivos fijados inicialmente, hace que la modularidad sea un factor importante en dicha herramienta, pudiendo realizarse cambios rápidos en la misma o adaptarla según nuevas funcionalidades, sin que sea necesario un gran esfuerzo para realizar dichos cambios.

Haciendo uso de esta metodología se deberán realizar los siguientes pasos para la obtención del producto final:

- Análisis de requisitos
- Planificación del trabajo
- Estudio de herramientas
- Implementación
- Verificación y pruebas del producto
- Revisión de cumplimiento de objetivos
- Cierre del proyecto y mantenimiento

1.1.5. Normativas

Este documento se ajusta a las normativas estructurales de documento para proyectos de Ingeniería Informática haciendo uso de las siguientes normativas:

- UNE 157001
- ISO/IEC/IEEE 16326

1.1.6. Estructura de la memoria

En este apartado se detallará el contenido global de este documento, explicando qué se encontrará en cada capítulo del mismo.

En el primer capítulo (Introducción) se encontrará la motivación de este proyecto, los objetivos a cumplir y el alcance que posee este TFG. En el capítulo siguiente (Conceptos preliminares y estado del arte), se explicarán conceptos necesarios para la comprensión del proyecto y el estilo de trabajo que se venía utilizando anterior al desarrollo de la herramienta.

El capítulo tercero (Desarrollo del calendario), desglosará las diferentes fases por las que pasará el proyecto e incluirá un diagrama Gantt para una visualización global de sus etapas. El siguiente capítulo (Descripción general del proyecto), contendrá la visión de los diferentes usuarios de la herramienta a la vez que se enumeran las funcionalidades de la misma. También contendrá las tecnologías y las herramientas seleccionadas para la realización del proyecto.

En los capítulos siguientes, (Análisis y Diseño), se mostrarán los diferentes casos de uso de la aplicación y el diseño por el que se ha optado a la hora de mostrar la información en la herramienta.

El capítulo octavo (Implementación), profundizará en la forma en la que actúa de manera interna la herramienta mostrando partes del desarrollo para tener una visión más clara de como realiza su trabajo. El capítulo noveno (Pruebas y validación), incluirá las pruebas pertinentes del sistema, para validar que todo funciona de la manera esperada y cumple los objetivos requeridos por el solicitante.

Finalmente, se incluye un capítulo que recopila el trabajo que se ha conseguido realizar y en qué podría mejorar la herramienta (Conclusiones y trabajo futuro).

Así mismo, este documento posee anexos donde se facilita información referente a la instalación del sistema (Manual de instalación), manuales para desarrolladores y usuarios (Manual de desarrollador) (Manual de usuario) además de una bibliografía (Bibliografía).

1.1.7. Glosario

El siguiente glosario contiene los acrónimos que aparecen a lo largo de este documento ordenados por su orden de aparición en el mismo.

1.1.7.1. Acrónimos

TFG Trabajo Fin de Grado

UCA Universidad de Cádiz

DNI Documento Nacional de Identidad

BD Base de Datos

XHTML eXtensible HyperText Markup Language

JS JavaScript

AJAX Asynchronous JavaScript And XML

SQL Structured Query Language

GIT Software para el control de versiones

CU Caso de uso

DS Diagrama de secuencia

IDE Integrated Development Environment (Entorno de desarrollo integrado)

LDAP Sistema centralizado de autenticación

JPA Java Persistence API

JSF JavaServer Faces

SQL Structured Query Language

MD5 Message-Digest Algorithm 5

Capítulo 2

Conceptos preliminares y estado del arte

Para la correcta comprensión del proyecto, es necesario el conocimiento de ciertos conceptos sobre la temática del mismo, las entidades implicadas y el estado en el que se encontraban estas previo al desarrollo de este TFG. En este capítulo se describirán dichos conceptos y situaciones.

2.1. Forma de trabajo actual

La gestión de la Hermandad se realiza mediante bases de datos locales y una aplicación de escritorio muy obsoleta dificultando así la disponibilidad del trabajo a realizar ya que, solo se podía trabajar en un solo equipo y por una sola persona. Esto además, conlleva un riesgo asumido de pérdida de datos ya que no se cuenta con ningún tipo de back-up automatizado que salve la información con la que se trabaja.

La gestión de la procesión se realiza mediante convocatoria a los hermanos para que se presenten en los días que se citan para cumplimentar su solicitud. Allí, in situ, el responsable de la Hermandad mediante un portátil y un formulario Word cumplimenta la solicitud pidiéndole información al hermano y, posteriormente, imprime un justificante y se le sella al hermano un papel que sirve como control de acceso a la iglesia para el día de salida.

Las túnicas se almacenan en cajas y solo tienen numeración. Al ser diferente los colores de túnicas de la Virgen y del Cristo, es necesario normalizar la numeración de la siguiente forma: Si es Virgen, la numeración empezará con **V-** y si fuese Cristo, la numeración empezaría de la siguiente forma: **X-**.

Además, no se tiene conocimiento del largo ni ancho de la túnica por lo que si un hermano necesita túnica nueva tiene que aguardar cierto tiempo en la sala probándose túnicas hasta dar con la suya.

Dichos archivos, llegaban a manos del Secretario para ser introducidos a la base de datos local y, según el caso, podían tener diferentes errores tales como:

- Error de caligrafías
- Inclusión de DNI en formato incorrecto.
- Falta de datos esenciales para ponerse en contacto con el hermano
- Datos sin cumplimentar por desconocimiento de su utilidad.
- etc...

Todo este tipo de errores debían ser depurados manualmente por el Secretario en los archivos formato Word citados anteriormente para poder ser aceptados en el sistema, lo que se convertía en una actividad ardua, poco eficiente y propensa a cometer errores de carácter humano, fallos de inconsistencia de la información con la BD local o fallos como los citados en los puntos anteriores.

2.2. Conclusión

Debido a todos los problemas que acarreaba el sistema de trabajo actual, como hermano de la cofradía que soy, estimé oportuno la creación de una herramienta que agilizara el trabajo a la par que controlase ciertos aspectos del tratamiento de la información para así minimizar los posibles errores en los mismos y consiguiendo que el factor error humano se redujese al mínimo posible.

De esa necesidad, surge la idea y motivación para este proyecto, ya que implica mejorar la calidad de trabajo de la hermandad, modernizando su trabajo diario y facilitando la vida a los hermanos que solicitan ser parte de la salida procesional. Al fin y al cabo, el patrimonio humano de dicha congregación es lo mas importante.

2.3. Investigación sobre herramientas

2.3.1. Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

Nació con el objetivo de ser un lenguaje de programación de estructura sencilla que pudiera ser ejecutado en diversos sistemas operativos. En lo que respecta a su nombre, inicialmente iba a denominarse Oak pero como esta marca ya estaba registrada se acabó optando por Java. Sirve para crear aplicaciones y procesos en una gran diversidad de dispositivos. Se base en programación orientada a objetivos, permite ejecutar un mismo programa en diversos sistemas operativos y ejecutar el código en sistemas remotos de manera segura.



Figura 2.1: Java

2.3.2. JPA

Java Persistence API, más conocida por sus siglas JPA, es la API de persistencia desarrollada para la plataforma Java EE. Es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard y Enterprise.

JPA (Java Persistence API) nos permite establecer una correlación entre una base de datos relacional y un sistema orientado a objetos. Esta correlación es llamada ORM (Object Relational Mapping), la cual genera anotaciones sobre Entidades. JPA establece una interface común que es implementada por un proveedor de persistencia de nuestra elección (TopLink, EclipseLink, Hibernate, entre otros).

Al fin y al cabo, es un framework para la creación de clases Java y relacionarlas con nuestras tablas de BD mediante anotaciones. Algunas anotaciones son las siguientes:

- @Entity
- @Table
- @Column
- @Id
- @Basic
- @Transient
- @Enumerated
- etc...

2.3.3. JSF

JavaServer Faces es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.



Figura 2.2: JSF

2.3.4. PrimeFaces

PrimeFaces es una biblioteca de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Algunas de sus principales características son las siguientes:

- Soporte nativo de Ajax
- Kit para crear aplicaciones web para móviles.
- Es compatible con otras librerías de componentes, como JBoss RichFaces
- Uso de javascript no intrusivo (no aparece en línea dentro de los elementos, sino dentro de un bloque <script>).

- Es un proyecto Open Source, activo y bastante estable entre versiones



Figura 2.3: PrimeFaces

2.3.5. Google Charts

Google Charts es una librería que genera gráficas facilitada por Google como otras tantas librerías open source que dicha empresa dispone al uso del desarrollador. La base de la librería se ejecuta mediante JavaScript y se representa mediante una etiqueta de HTML.



Figura 2.4: Google Charts

2.4. Conclusión

Debido a que este proyecto consiste en una herramienta web para la gestión de una hermandad con un volumen alto de usuarios en el sistema, se hacía necesario una implementación que se adaptase al modelo de trabajo del desarrollo web y que estuviese enfocado a una herramienta accesible, sencilla y rápida, considerando un lenguaje universalmente conocido para el posible mantenimiento de la misma por otros equipos de desarrollo. Además, el muy posible crecimiento de la misma derivando así en una arquitectura que, perfectamente, puede ser asumida por un desarrollo Java por lo que todos estos requisitos hacen como mejor opción el desarrollo en dicho lenguaje.

Capítulo 3

Desarrollo del calendario

En este capítulo se describirán las etapas por las que el proyecto pasará desde la concepción del mismo hasta su implementación y puesta en marcha, adicionalmente, se incluye un diagrama Gantt que ilustra todas las fases que conforman el proyecto y un estudio sobre la viabilidad del mismo a través de los cálculos de costes para este proyecto.

3.1. Fases

Para la realización de este proyecto se ha desglosado su desarrollo en diferentes etapas que cubren todo el proceso de creación de la herramienta, desde su concepción teórica hasta su conclusión final tras la finalización del proyecto.

3.1.1. Fase primera: Inicio proyecto

El proyecto surge de la necesidad tener una herramienta de control y gestión de la Hermandad. Actualmente es complicada la labor ya que todo se realiza en papel con todo lo que ello conlleva: ortografía no muy clara, pérdida de documentos, necesidad de sello para todas las acciones, disponibilidad de impresora...etc. por ello, me puse en contacto con el Hermano Mayor de la Hermandad, que a ojos de la Hermandad, es la persona encargada de gestionarla en su totalidad y le propuse realizar la herramienta con el fin de proporcionarles un software de calidad, claro, robusto y que permite realizar las labores que ellos necesitan gestionar.

Así pues, me puse en contacto con mi tutor de mi Trabajo de Fin de Grado y le comenté la idea de desarrollar dicha herramienta para poder obtener mi Grado en Ingeniería Informática.

3.1.2. Fase segunda: Trabajo de investigación

El trabajo de investigación está basado en las tecnologías Java demandadas en nuestro entorno laboral. En Andalucía, un desarrollador Java tanto para la parte de Backend como la parte de Frontend es necesario que tenga conocimientos de JPA, JSF, Hibernate, Oracle... entre otras tecnologías para poder optar a un puesto de desarrollador de calidad. En mi actual trabajo, un proyecto para la Junta de Andalucía en el cual ejerzo una categoría de desarrollador senior, optamos por dichas tecnologías por lo que el trabajo de investigación y profundización en la materia se ha realizado en horario laboral y basándome en modelos de desarrollo ya existentes en el mundo real.

3.1.3. Fase tercera: Objetivos

Con la formación necesaria para la comprensión de la magnitud del proyecto y tras una reunión inicial con el solicitante, se fijaron unos objetivos prioritarios y funcionalidades requeridos en la herramienta. Como objetivos iniciales se pactaron los siguientes:

▪ Apartados de la herramienta:

- La aplicación deberá poseer apartados diferenciados para la gestión de **Hermanos, Controles de Salida, y Túnicas**.
- El apartado de Hermanos deberá ser capaz de listar los hermanos, añadir un nuevo hermano, dar de baja un hermano, o poder editarlo.
- El apartado de Túnicas deberá ser capaz de listar las túnicas, añadir una nueva túnica, dar de baja una, o poder editarla.
- El apartado de Control de Salida deberá ser capaz de listar los Controles diferenciando entre los hermanos que participan en la procesión de Cristo o procesión de Virgen, dar un número total de hermanos participante, dar un control de salida nuevo, eliminar un control de salida o poder editar uno.

▪ Perfiles:

- Para el objetivo inicial, se concreta tener un perfil de Administrador que será capaz de manejar la totalidad de la herramienta y será el encargado de la creación de los datos del sistema y de la administración ya que se considera que es una aplicación interna de la Hermandad.

Estos objetivos fueron incrementales durante el desarrollo de la herramienta, pues cada cierto tiempo, el prototipo era mostrado a la parte solicitante para comprobar si cumplía con las necesidades pactadas inicialmente o, por el contrario, requería de algún cambio, mejora o nueva funcionalidad. De esta forma, los objetivos inicialmente fijados necesitaron de ciertas revisiones o un re-enfoque a medida que el producto iba siendo implementado.

Tras reuniones con la parte solicitante, se indicaron nuevas funcionalidades y exigencias para con la herramienta, incluyendo sugerencias del desarrollador como opciones de mejora y que aportaban mayor facilidad en cuanto al uso de la herramienta.

Finalmente los objetivos a cumplir fueron los siguientes:

▪ Apartados de la herramienta:

- La aplicación deberá poseer apartados diferenciados para la gestión de **Hermanos, Controles de Salida, Túnicas y Personas**.
- El apartado de Hermanos deberá ser capaz de listar los hermanos, añadir un nuevo hermano, dar de baja un hermano, o poder editarlo.
- El apartado de Túnicas deberá ser capaz de listar las túnicas, añadir una nueva túnica, dar de baja una, o poder editarla.
- El apartado de Control de Salida deberá ser capaz de listar los Controles diferenciando entre los hermanos que participan en la procesión de Cristo o procesión de Virgen, dar un número total de hermanos participante, dar un control de salida nuevo, eliminar un control de salida o poder editar uno.
- El apartado de Personas deberá ser capaz de listar las personas pertenecientes a la aplicación así como dar de alta, baja o editar una.
- Poder generar un documento PDF con el justificante del control de salida del hermano para que tenga, de forma digital, su recordatorio en papel con un diseño cuidado y elegante.
- Poder enviar correos, en el caso de que el hermano haya facilitado su correo electrónico, al hermano siempre que se genere o edite su control de salida para que siempre tenga en su correo el justificante y la acción que se ha realizado y sirva como justificante el día de la salida procesional evitando así pérdidas u olvidos del justificante en papel.
- A la aplicación deberá poseer apartados diferenciados para la gestión de **Tus cuotas y Reserva tu control de salida**.

- En el apartado de Tus Cuotas, accesible desde un rol de Usuario, el hermano podrá obtener información de las cuotas pertenecientes a su ficha así como la posibilidad de cambiar su modalidad de pago ofreciendo las siguientes opciones:
 - Metálico
 - Transferencia Bancaria
 - PayPal
 - Generación de Estadísticas como herramienta interna de la aplicación.
 - En el apartado de Reserva tu control de salida, accesible desde un rol de Usuario, el hermano podrá formalizar su participación en la procesión cumplimentando un formulario facilitado por la aplicación web en el que se ofrecen también diferentes opciones de pago:
 - Metálico
 - Transferencia Bancaria
 - PayPal
 - Para el apartado Reserva tu control de salida, el hermano puede cumplimentar el formulario pero no puede seleccionar su ocupación en la procesión por lo que se genera un aviso en forma de correo cuando la Hermandad modifica su salida procesional y le asigna tanto ocupación como sección en la procesión.
- **Perfiles:**
- La aplicación debe implementar dos roles para hacer uso de la herramienta. Dichos roles son: **Administrador** y **Usuario**.
 - Para el rol **Administrador**, la aplicación mostrará los apartados de **Hermanos**, **Controles de Salida**, **Túnicas** y **Personas**.
 - Para el rol **Usuario**, la aplicación mostrará los apartados de **Tus cuotas** y **Reserva tu control de salida**.

3.1.4. Fase cuarta: Desarrollo

Una vez fijados los objetivos a realizar, se comenzó la implementación de la herramienta, usando un modelo de desarrollo en cascada, aplicándolo sobre cada apartado de la aplicación por lo que a medida que se iba desarrollando... iban surgiendo nuevas funcionalidades para la aplicación y había que volver a estimar y analizar el impacto de los mismos sobre lo existente.

Durante el tiempo de desarrollo se fueron realizando diferentes versiones del manual de uso de dicha aplicación, de cara a facilitar el uso de la herramienta a los usuarios finales, desglosando cada apartado de la herramienta en función del usuario accediese a la misma. Dicho manual está incluido en el anexo “Manual de Usuario” del presente documento.

Todo el desarrollo de la herramienta viene ampliado en el capítulo correspondiente en este documento. (Capítulos 4, 5, 6 y 7).

3.1.5. Fase quinta: Pruebas y depuración

Finalizado el periodo de desarrollo de la herramienta, se realizaron las pruebas necesaria para comprobar el buen funcionamiento de la aplicación. Estas pruebas también se realizaron a medida que el desarrollo iba incrementando, para así constatar que el trabajo realizado no contenía fallos de consistencia para con la información de las bases de datos en juego de la herramienta.

Estas pruebas, recogidas en el capítulo “Pruebas y validación” de este documento, se centran en pruebas unitarias y de funcionamiento, así como en un exhaustivo control a la hora del desarrollo,

como lo son la reutilización de código, comentarios pertinentes para desarrolladores y búsqueda de claridad en el mismo.

3.1.6. Fase sexta: Conclusiones

Llegados a esta fase, con el proyecto concluido, se validaron los objetivos a cumplir en la herramienta para comprobar si se han cumplido las expectativas respecto al proyecto. Además se presenta en modo local la aplicación al cliente. En este caso, la conclusión del mismo es positiva y se determina la posibilidad de subirlo a un hosting a convenir para poder usarla y tener unos meses la aplicación con un mantenimiento.

3.2. Costes

Los costes para este proyecto se encuentran desglosados en dos categorías, costes de recursos humanos, donde se indicarán los valores del esfuerzo a la hora del desarrollo de la herramienta, así como la mano de hora para el desarrollador, y costes de recursos materiales, como son las licencias en uso para el desarrollo, equipos que hacen posible la puesta en marcha de la aplicación y su posterior mantenimiento, y los derivados para el correcto funcionamiento de la misma.

3.2.1. Costes de recursos humanos

Para la estimación de costes en recursos humanos se ha establecido un salario acorde con un perfil de graduado en Ingeniería Informática, donde, realizando una consulta estadística sobre el valor en el mercado de hora/trabajo en el campo del desarrollo web, se estimó una contribución monetaria de 30€/h.

Fijado dicho precio, calcularemos el coste de recursos humanos, estimando el tiempo de desarrollo para la herramienta que será de 2 meses y una semana con un trabajo semanal de 10 horas semanales. La siguiente tabla contiene el desglose monetario citado anteriormente a modo de resumen:

Salario/Hora	Horas Realizadas	Total
30€/h	90hs	2700€

Tabla 3.1: Costes de Recursos Humanos

3.2.2. Costes de recursos materiales

En los recursos materiales se enumerarán todas las herramientas utilizadas para el desarrollo web de una aplicación, así como las licencias y costes derivados del uso de servicios que se suministren para el buen funcionamiento de la herramienta desarrollada en este proyecto.

Cabe destacar que para la realización de este TFG se procurará el uso de herramientas liberadas, de código abierto y gratuitas en la medida de lo posible, desarrollando el trabajo bajo el sistema operativo iOS. La siguiente tabla contiene un desglose monetario de los programas y licencias que se usarán para el desarrollo de la herramienta:

Producto/Licencia	Precio	Total
Eclipse	Gratuito	0€
MySQL	Gratuito	0€
PrimeFaces	Gratuito	0€
GitHub	Gratuito	0€
Editor de texto LaTeX	Gratuito (ShareLatex)	0€
Editor Gantt	Gratuito (GanttProyect)	0€

Tabla 3.2: Costes de Recursos Materiales

Todas las herramientas citadas en la tabla anterior, serán desglosadas y se indicará el por qué de su uso en la sección correspondiente de este documento, Capítulo 4, apartado Herramientas.

3.3. Diagrama de Gantt

Se incluye un diagrama Gantt que reúne las fases mostradas anteriormente a modo de gráfico donde se sitúan las fechas y duración de cada etapa, así como, el tiempo estimado de la realización del proyecto completo, contando con la documentación del mismo.

Para una mejor visualización del gráfico Gantt se facilita en una página independiente del documento y en formato vertical. Lo podrá encontrar en la página siguiente.

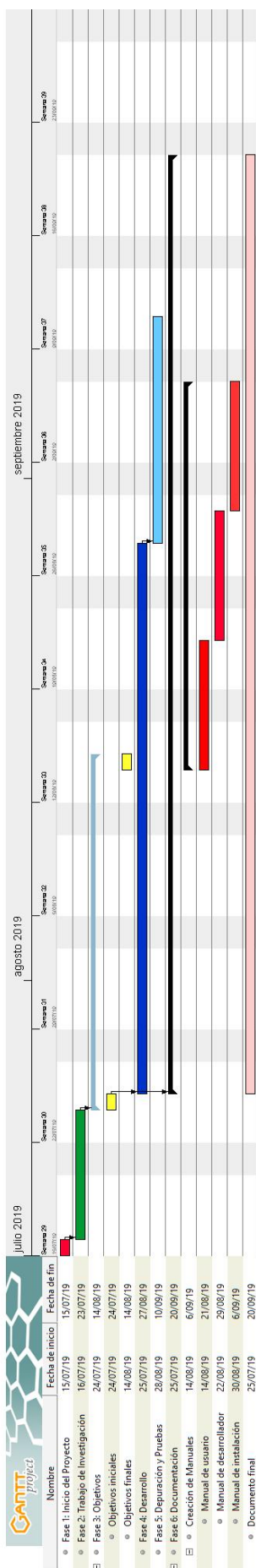


Figura 3.1: Diagrama Gannt.

Capítulo 4

Descripción general del proyecto

En este capítulo se desarrollarán los diferentes puntos de visualización que posee la aplicación, los módulos y funcionalidades que posee la misma, así como, las diferentes tecnologías y lenguajes usados para su creación.

4.1. Descripción del proyecto

El proyecto TUNICAPP: TU APLICACIÓN WEB PARA LA GESTIÓN DE TU HERMANDAD realizará la gestión interna de la Hermandad y la gestión pública del Hermano. Para ello, la aplicación cuenta con diferentes apartados o secciones que facilita la labor al responsable de la gestión interna tales como dar de alta un Hermano, asignar un tutor legal para un hermano menor de edad, gestionar las túnicas disponibles para las salidas procesionales, informar al hermano mediante correo electrónico de los trámites que se están realizando, gestionar y organizar la salida procesional... entre otras opciones.

Por otro lado, los apartados o secciones públicas pertenecientes al Hermano pueden ser gestionar sus cuotas, ver el estado del pago, la descripción de su ficha así como su situación en la salida procesional si la tuviese. Además, para facilitar la labor de la Hermandad en las citaciones anuales donde se forman grandes colas para tramitar el control de salida existe la posibilidad de tramitarla de forma online por la plataforma ahorrando tiempo, disminuyendo el error del factor humano y evitando pérdidas de documentos. Se implementa además la posibilidad de pago mediante transferencia bancaria o mediante PayPal.

Se incorpora una obtención de gráficas basada en las estadísticas internas de la aplicación como forma de reflejar información relevante para el usuario.

4.2. Perspectiva del proyecto

Debido a la necesidad de la herramienta de poseer diferentes tipos de usuarios, se hace necesario la creación de diversas formas de mostrar la información para según qué usuario esté haciendo uso de la misma.

Esta forma de mostrar la herramienta no solo se basa en el apartado visual, si no que además, dota a la aplicación de una seguridad extra frente a accesos no autorizados por parte de los usuarios que no posean los roles y permisos destinados para el uso de los diferentes módulos de la herramienta.

En un apartado posterior de este capítulo se indicará qué perfil tiene acceso a según qué acciones dentro un módulo, poseyendo diferentes visualizaciones de la misma u opciones diferentes según el rol que posea el usuario.

En función a este concepto, la herramienta es construida en base a módulos donde el usuario podrá realizar las acciones solicitadas por el cliente indicadas en las diferentes fases del capítulo anterior

de este documento.

4.3. Metodología de desarrollo

En este proyecto se ha usado una metodología de desarrollo de **desarrollo en cascada**.

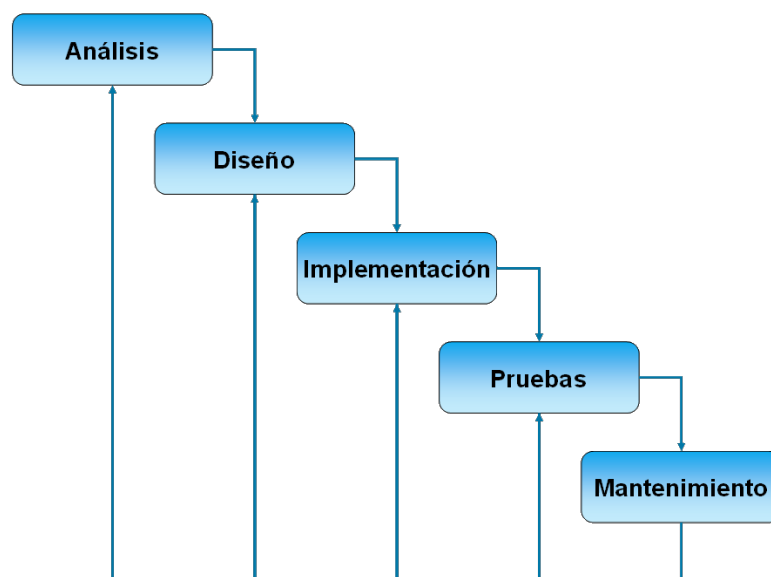


Figura 4.1: Modelo en cascada.

El desarrollo en cascada, (denominado así por la posición de las fases en el desarrollo de ésta, que parecen caer en cascada “por gravedad” hacia las siguientes fases), es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase.

El ciclo de vida de esta metodología abarca las siguientes actividades:

- **Análisis de requisitos** En esta fase son analizadas las necesidades de los usuarios finales del producto, para poder determinar que objetivos se han de cubrir. Como resultado de esto se obtiene una memoria, llamada **documento de especificación de requisitos** que recoge la especificación de lo que debe hacer el sistema sin entrar en demasiados detalles internos. En esta etapa se debe consensuar todo lo que se necesita del sistema y será lo que se seguirá en las siguientes fases, sin poderse requerir nuevos resultados a la mitad del proceso de elaboración del software.
- **Diseño del sistema** Divide el sistema en elementos que se pueden realizar por separado, aprovechando las ventajas del desarrollo en equipo. Se obtiene tras esto el **documento de diseño del software**, que describe la estructura relacional global del sistema y la especificación del cometido de cada una de sus partes, además de la manera en la que éstas se combinan. Se puede distinguir entre diseño de alto nivel, donde se define la estructura de la solución y sus relaciones, y el diseño detallado, donde se definen los algoritmos empleados y la organización del código para comenzar a implementar.
- **Diseño del programa** En esta etapa se realizan los algoritmos necesarios para que se cumplan los requisitos del usuario, además de los análisis necesarios para saber qué herramientas utilizar en la siguiente fase.

- **Desarrollo** Es la etapa en la que el código fuente se implementa, usando tanto prototipos como pruebas para corregir los errores que puedan aparecer.
- **Pruebas** Una vez programados los elementos, se unen para componer el sistema y se comprueba que funcionen correctamente cumpliendo los requisitos establecidos, antes de ser entregado al usuario final.
- **Verificación** En esta fase el usuario final ejecuta el sistema, habiéndose realizado las pruebas necesarias para que el sistema no falle.
- **Mantenimiento** Esta etapa es una de las más críticas. Se trata del mantenimiento del software, ya que puede ser que una vez utilizado por el usuario final, no cumpla todas sus expectativas.

Esta metodología tiene una serie de **ventajas y desventajas**, que son las siguientes:

- **Ventajas**
 1. Buen funcionamiento en productos maduros. Hacen falta menos herramientas para que funcione de forma óptima.
 2. Orientado a documentos.
 3. Entendible y fácil de implementar.
 4. Metodología de trabajo efectiva: definir antes que diseñar y diseñar antes que codificar.
- **Desventajas**
 1. Un proyecto no suele seguir una secuencia lineal en la vida real. Ésto conlleva a una mala implementación del modelo.
 2. Largo proceso de creación del software porque debe pasar por el proceso de prueba, lo que hace que no se opere hasta que el proyecto esté completo.
 3. Cualquier error de diseño que se detecte en la fase de prueba hace que se rediseñe el software y se programe de nuevo el código afectado. Ésto aumenta los costes de desarrollo.
 4. Una etapa no se puede comenzar hasta haber acabado la anterior.

En definitiva, he utilizado esta metodología por las siguientes razones:

1. Al ser secuencial, es mejor para un proyecto que realice una persona, ya que al tener una gran planificación, quedan muy claros los pasos que hay que realizar y no se pasa nada por alto.
2. Por tener esa gran planificación, se puede iniciar el software con bastante rapidez.
3. Puede estimar plazos y costes de manera más precisa, algo que complace a los clientes.

4.4. Módulos y Funcionalidades

Previo a mostrar la herramienta en la visión particular de cada perfil, se hace necesario desglosar los apartados o módulos que contiene la herramienta de cara a una mejor comprensión de la misma.

Para ello, el usuario deberá iniciar sesión en la herramienta y se mostrará la pantalla de inicio de la aplicación.

Ciertas acciones de la herramienta tan solo son accesibles para algunos tipos de perfil dentro de la misma, con lo que, para un visualizado general de la aplicación, se mostrarán los módulos de forma individualizada ya que, en esta herramienta, el perfil Administrador no contiene las vistas del perfil Hermano.

4.4.1. Inicio

Una vez dentro de la aplicación y como se comentaba en la introducción de las funciones, la aplicación se desglosa en diferentes módulos, donde en cada apartado de la herramienta se realizan las acciones necesarias para cada una de las funcionalidades requeridas.

En este módulo se mostrará una pequeña bienvenida al sistema y se accederá al sistema haciendo uso del usuario y contraseña que posea. De aquí en adelante, la información de forma detallada se encuentra en el manual de Usuario.

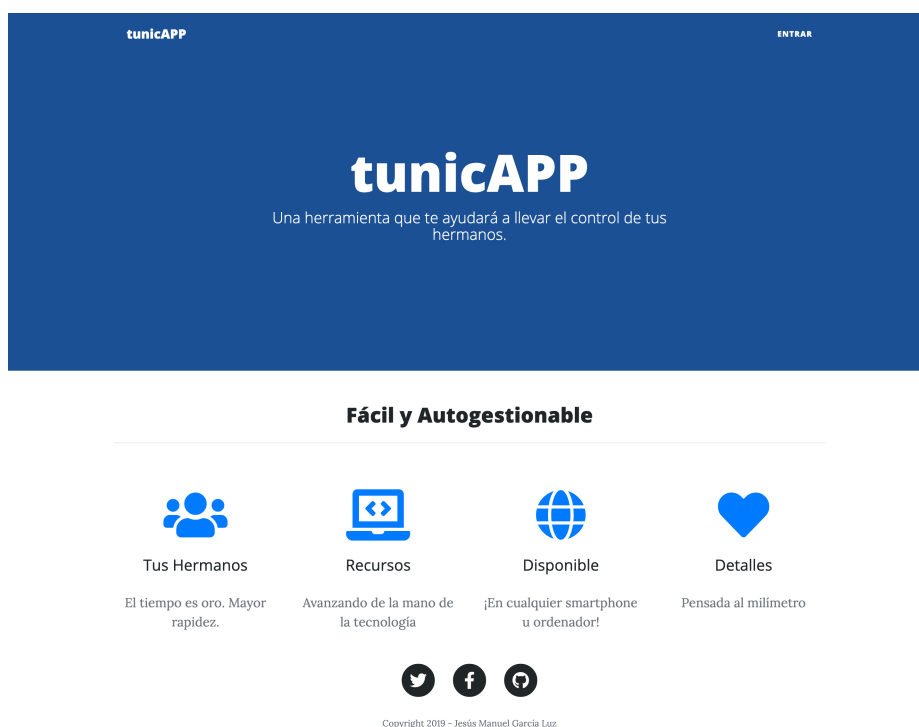


Figura 4.2: Bienvenid@ a tunicAPP.

El manual puede encontrarse en los anexos del presente documento en el capítulo correspondiente facilitado para ello (Anexo C: Manual de usuario).

4.4.2. Inicio de Sesión

En el módulo de sesión podremos introducir nuestras claves facilitadas bien por el creador del sistema (en el caso del administrador) o bien mediante correo electrónico. Además, se cuenta con una recuperación de claves la cual resetea la contraseña y se le envía una nueva. El administrador jamás sabrá la clave usada con anterioridad. Cabe destacar que la clave está cifrada en MD5 y que hace altamente difícil su hackeo.

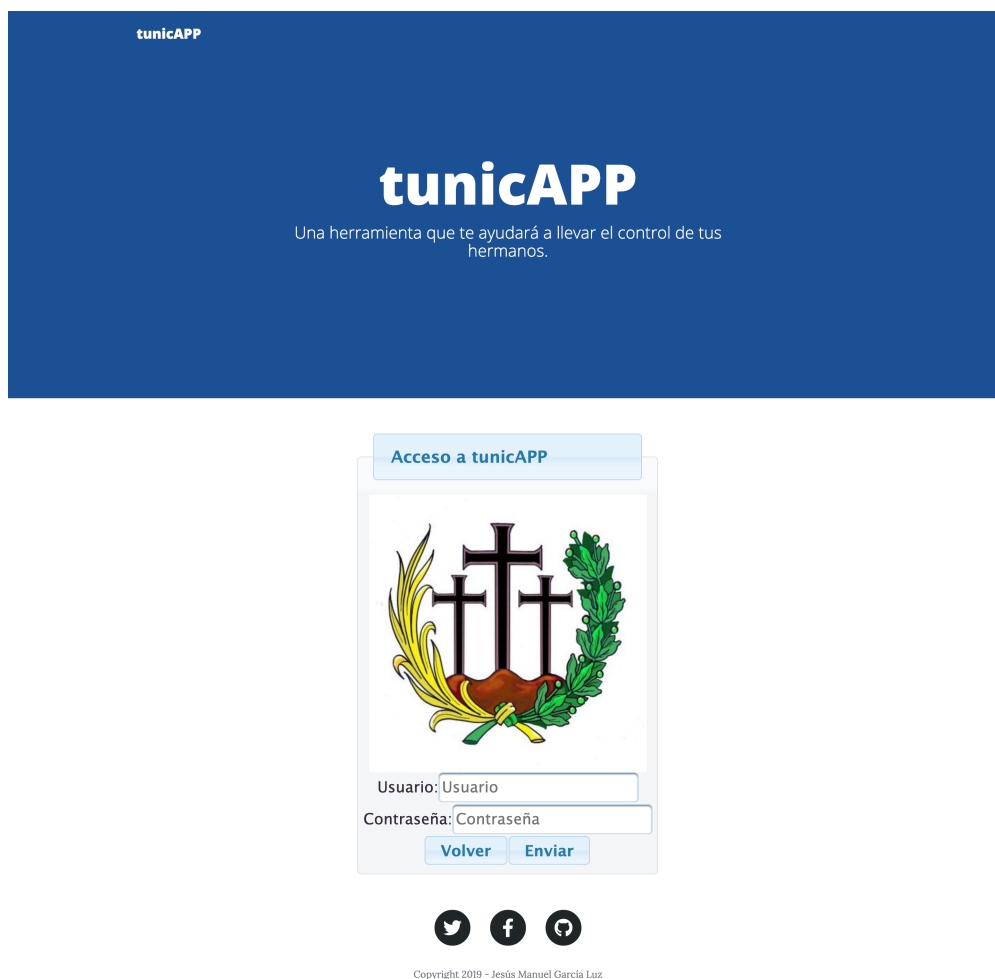


Figura 4.3: Pantalla de login.

Una vez realizado el inicio de sesión, entraremos en vistas diferentes. En el caso del Administrador podremos ver los siguientes apartados troncales:

- Personas
- Hermanos
- Control de Salida
- Túnicas

Cada uno de ellos, se subdividen en otros menús y que hablaremos de ellos en este capítulo.

Por parte del Hermano obtenemos los siguientes módulos troncales:

- Tus cuotas
- Reserva tu control de Salida


4.4.3. Módulos del Administrador

4.4.3.1. Personas

En el módulo de Personas, el administrador del sistema podrá dar de alta, baja, editar y eliminar Personas. Cabe destacar que en nuestro sistema, las personas puede que no sean hermanos, sin embargo, los hermanos están obligados a ser entidades Persona. Esto se debe a que en el sistema se podría requerir en un futuro dar de alta a Hermanos menores de edad que no tengan documento nacional de identidad, y que, a la hora de dar el alta, dichas personas actúen de Autorizadas y responsables del mismo. Los subniveles asociados al módulo de Personas son los siguientes:

- **Lista de Personas**

En este módulo obtendremos la lista total de personas dadas de alta en el sistema además de datos que se muestran de forma clara y concisa en una tabla para poder hacer uso de la información en el momento de consulta. Además, dicha tabla posee la posibilidad de exportar el listado a Excel o a PDF facilitando así su exportación y trabajo fuera del sistema. Esta utilidad es muy útil para el administrador para suministrar a sus ayudantes de los listados pertinentes.



tunicAPP
Una herramienta que te ayudará a llevar el control de tus hermanos.

Listado de Personas

DNI	Nombre	Primer Apellido	Segundo Apellido	Telefono	Email	Dirección
76647220L	Jesus Manuel	Garcia	Luz	677117463	yisekai@gmail	Calle Jorge de Montemayor 5, 2D
03788967Q	David	Garcia	Luz	666666666	yisekai@gmail	Calle Granada 4, Piso 4 Puerta 2
30536271E	Bea	Rodriguez		666666666	yisekai@gmail	Calle Esperanza de la Trinidad
30263890F	Beatriz	Rubio	Rodriguez	609143262	rubiorbeatriz	Esperanza Trinidad 5 bloqueada 11 1c
30263891P	Alberto	Rubio	Rodriguez	650637377	rubiorbeatriz	Esperanza trinidad 5 bloq 11 1c
Y0499240Q	Persona	Extranjera		677117463	yisekai@gmail	Calle Granada 4

Copyright 2019 - Jesús Manuel García Luz

Figura 4.4: Listado de Personas.

■ Añadir Persona

En este módulo el administrador del sistema facilita un formulario de alta de una persona. Entre los datos que se solicitan, existe un combo para el tipo de documento de identidad, el número de documento, el nombre, primer apellido, segundo apellido, teléfono, email, fecha de nacimiento y domicilio. Además, el sistema le solicita a la persona que se va a registrar autorizar a la hermandad y al sistema hacer uso de los datos personales mostrando y pudiendo marcar en un check dicha autorización para la LOPD. Sin dicha autorización no es posible dar de alta a la persona en el sistema.

The image shows the 'tunicAPP' web application interface. At the top, there is a navigation bar with the logo 'tunicAPP' and several menu items: 'PERSONAS', 'CONTROL DE SALIDA', 'HERMANOS', and 'TÚNICAS'. Below the navigation bar, the main header area features the 'tunicAPP' logo and the tagline 'Una herramienta que te ayudará a llevar el control de tus hermanos.'.

The main content area is titled 'Alta' (Registration). It contains a 'Formulario' (Form) section with the following fields:

- DNI/NIE (*)**: A dropdown menu with 'Seleccione' as the current selection.
- Número de Documento (*)**: A text input field.
- Nombre (*)**: A text input field.
- Primer Apellido (*)**: A text input field.
- Segundo Apellido**: A text input field.
- Teléfono (*)**: A text input field.
- Email**: A text input field.
- Fecha de Nacimiento (*)**: A date picker with 'dia/mes/año' format.
- Domicilio (*)**: A text input field.

Below the form fields, there is a checkbox and a paragraph of text:

☐ Autorizo a la Venerable Cofradía de Penitencia del Santísimo Cristo del Perdón y María Santísima del Rosario en sus Misterios Dolorosos al tratamiento de mis datos personales (incluyendo su incorporación en los correspondientes ficheros) incorporados en la solicitud y en otros documentos normalizados relativos a la acción formativa, tal y como se dispone en la Ley Orgánica 15/1999, de 13 de diciembre de Protección de Datos de Carácter Personal (LOPD), para su tratamiento o consulta en las acciones que dicha corporación vea oportuna.

At the bottom of the form is an 'Enviar' (Send) button.

Below the form, there are three social media icons: Twitter, Facebook, and YouTube. At the very bottom, there is a copyright notice: 'Copyright 2010 - Jesús Manuel García Láz'.

Figura 4.5: Alta de Personas

■ Editar Persona

En este módulo el administrador del sistema ofrece una pantalla de búsqueda de la persona buscándola mediante un DNI facilitado. El sistema hará la búsqueda del DNI y ofrecerá una vista donde aparezcan los campos a editar de la persona en el caso de que la encuentre.

tunicAPP

PERSONAS · CONTROL DE SALIDA · HERMANOS · TÚNICAS

tunicAPP

Una herramienta que te ayudará a llevar el control de tus hermanos.

Editar Persona

Panel de Búsqueda

DNI/NIE
76647220L

Enviar Limpiar

Datos de Jesus Manuel Garcia Luz:

DNI/NIE
DNI

Número de Documento
76647220L

Nombre
Jesus Manuel

Primer Apellido
Garcia

Segundo Apellido
Luz

Teléfono
677117463

Email
yisekai@gmail.com

Fecha de Nacimiento
04/03/1991

Domicilio
Calle Jorge de Montemayor 5, 2D

☒

Autorizo a la Venerable Cofradía de Penitencia del Santísimo Cristo del Perdón y María Santísima del Rosario en sus Misterios Dolorosos al tratamiento de mis datos personales (incluyendo su incorporación en los correspondientes ficheros) incorporados en la solicitud y en otros documentos normalizados relativos a la acción formativa, tal y como se dispone en la Ley Orgánica 15/1999, de 13 de diciembre de Protección de Datos de Carácter Personal (LOPD), para su tratamiento o consulta en las acciones que dicha corporación vea oportuna.

Editar

Twitter Facebook GitHub

Copyright 2019 - Jesús Manuel García Luz

Figura 4.6: Editar Persona.

■ Borrar Persona

En este módulo el administrador del sistema ofrece una pantalla de búsqueda de la persona buscándola mediante un DNI facilitado. El sistema hará la búsqueda del DNI y borrará a la persona de nuestra base de datos mostrando un mensaje satisfactorio en caso positivo.

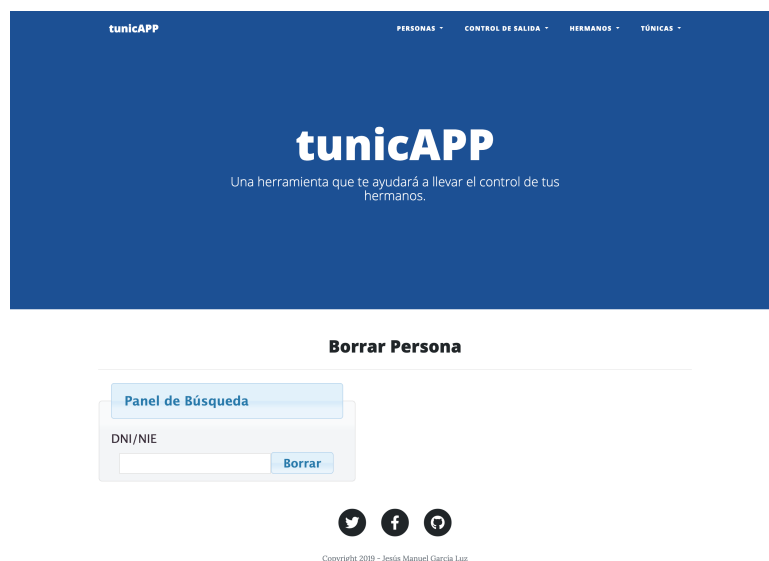


Figura 4.7: Borrar Persona.

4.4.3.2. Hermanos

En el módulo de Hermanos, el administrador del sistema podrá dar de alta, baja, editar y eliminar Hermanos. Los hermanos es el núcleo de nuestra aplicación. Toda la hermandad se rige en base a dicha entidad ya que es la pieza orquestadora de las otras entidades. Ser hermano conlleva pagar un recibo anual.

Los subniveles asociados al módulo de Hermanos son los siguientes:

- **Lista de Hermanos**

En este módulo obtendremos la lista total de hermanos dadas de alta en el sistema además de datos que se muestran de forma clara y concisa en una tabla para poder hacer uso de la información en el momento de consulta. Por otro lado, dicha tabla ofrece la posibilidad de exportar el listado a Excel o a PDF facilitando así su exportación y trabajo fuera del sistema. Esta utilidad es muy útil por el administrador para suministrar a sus ayudantes de los listados pertinentes.

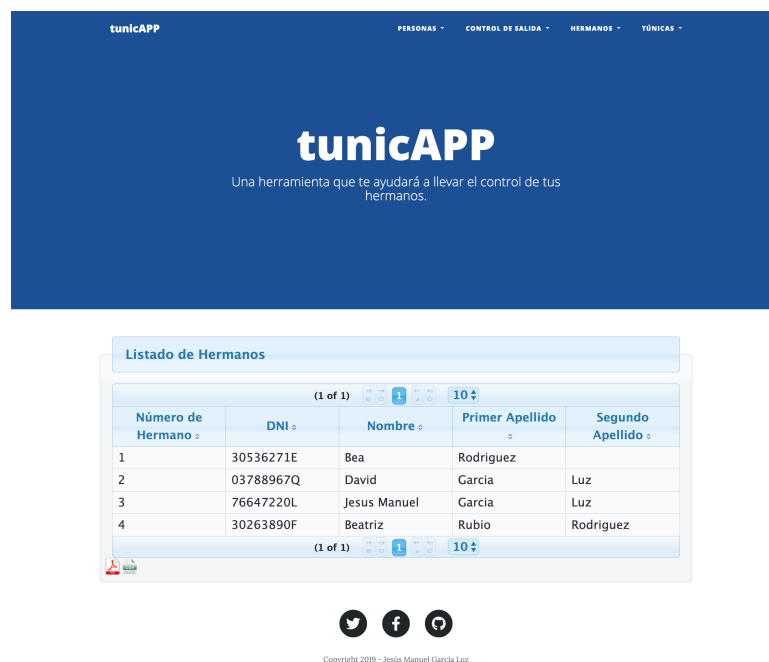


Figura 4.8: Listado de Hermanos.

■ Añadir Hermano

Como hemos visto en el módulo anterior un hermano previamente debe ser una Persona así que, el sistema para hacer uso de este módulo debe introducir el DNI de una persona del sistema a la cual quiera registrar como hermano.

Al buscar a la persona, el sistema traerá precargado los datos de la Persona y podrá añadir un recibo de Hermano. Dicho recibo es configurable en Tipo de Pago (metálico, PayPal o Transferencia Bancaria) como en número de cuotas.

■ Editar Hermano

El sistema mostrará una pantalla de búsqueda mediante DNI del Hermano y ofrecerá al usuario de la aplicación los campos posibles a editar.



Figura 4.9: Añadir Hermano

■ Borrar Hermano

En este módulo el administrador del sistema ofrece una pantalla de búsqueda de la persona mediante un DNI facilitado. El sistema hará la búsqueda del DNI y borrará al hermano asociado de nuestra base de datos mostrando un mensaje satisfactorio en caso positivo.



Figura 4.10: Borrar Hermano

4.4.3.3. Control de Salida

En el módulo de Control de salida el administrador del sistema podrá dar de alta, baja, editar y eliminar Controles de Salida. Los controles de salidas son los trámites a realizar por el hermano para poder participar en la salida procesional. Dichos trámites conllevan el pago de un recibo como donativo y le da derecho, si el hermano quisiera, de participar en una sección de la procesión portando un cirio o una insignia a decidir por la Hermandad.

Los subniveles asociados al módulo de Control de Salida son los siguientes:

- **Lista de Controles de Salida**

En este módulo obtendremos la lista total de controles de salida dadas de alta en el sistema además de datos que se muestran de forma clara y concisa en una tabla para poder hacer uso de la información en el momento de consulta. La información se muestra dividida por el paso al que pertenecen ya que un control de salida está designado al paso del Cristo o de la Virgen. Para contabilizar ambos, tenemos dos tablas individuales que facilita la labor.

Por otro lado, dichas tablas ofrecen la posibilidad de exportar los listados a Excel o a PDF facilitando así su exportación y trabajo fuera del sistema. Esta utilidad es muy útil por el administrador para suministrar a sus ayudantes de los listados pertinentes.

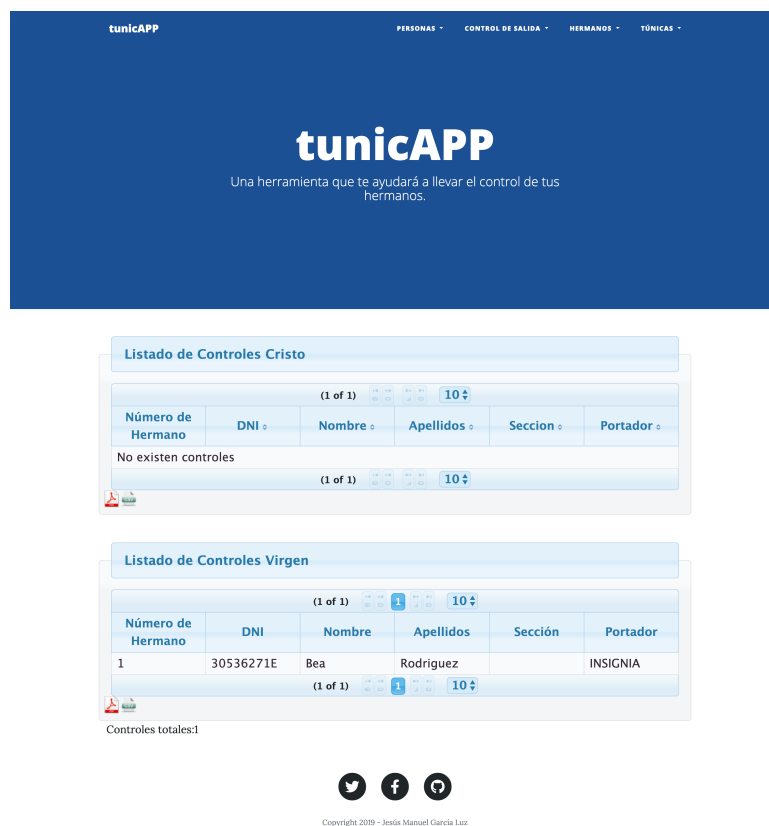


Figura 4.11: Listado de Controles de Salida.

■ Añadir Control de Salida

Como hemos visto en módulos anteriores, todo se rige mediante la búsqueda del hermano/-persona en concreto mediante su DNI. Posteriormente se muestran los datos precargados del Hermano y se ofrece la posibilidad de dar de alta un recibo de salida procesional. En esta sección solo podemos asignar el tipo de pago como metálico ya que el administrador no tiene poder para pagar el recibo de otra persona, no obstante, lo puede marcar como pagado ya que en la mayoría de los casos los hermanos pagan 'in situ' su control de salida mientras le realizan el formulario.

Para participar en la procesión es necesario asignarle túnica al hermano que, o bien puede traerla precargada del año anterior, o bien hay que asignar una nueva. El módulo de túnicas se describirá mas adelante en esta misma sección.

The screenshot displays the 'tunicAPP' web interface. At the top, there is a navigation bar with links for 'PERSONAS', 'CONTROL DE SALIDA', 'HERMANOS', and 'TÚNICAS'. The main header features the 'tunicAPP' logo and the tagline 'Una herramienta que te ayudará a llevar el control de tus hermanos.' Below this, the title 'Alta Control de Salida' is centered. The form is divided into several sections: 1. 'Panel de Búsqueda' with a 'DNI/NIE (*)' field containing '76647220L' and 'Buscar'/'Limpiar' buttons. 2. 'Datos de Jesus Manuel Garcia Luz:' showing 'DNI/NIE' as 'DNI' with value '76647220L', 'Número de Documento' as '76647220L', 'Nombre' as 'Jesus Manuel', and 'Apellidos' as 'Garcia Luz'. 3. 'Control de Salida' with 'Tipo de Recibo (*)' set to 'Salida Procesional 2019', 'Cuantía' set to '20', and 'Pagado' set to 'No', with an 'Establecer como pagado' button. 4. 'Túnica' section with 'Titular (*)' and 'Portador (*)' both set to 'Seleccione', and 'Túnica (*)' set to 'Seleccione'. At the bottom, there is a 'Dar de Alta el Control' button, social media icons for Twitter, Facebook, and Instagram, and a copyright notice: 'Copyright 2019 - Jesus Manuel Garcia Luz'.

Figura 4.12: Añadir Control de Salida

■ Editar Control de Salida

El sistema mostrará una pantalla de búsqueda mediante DNI del Hermano y ofrecerá al usuario de la aplicación los campos posibles a editar.

Una vez realizada la búsqueda, como datos a destacar para modificar son 'Marcar como pagado' que actualiza el recibo de la salida procesional a pagado, la modificación de la túnica previa entrega de la anterior asignada, y la adjudicación de la sección e insignia.

A los hermanos, cuando se les hace el control de salida no se le dice en que sección irán ni que llevarán en la procesión hasta que acabe la fecha de inscripción a la salida procesional. Una vez concluido el plazo, la cofradía se reúne y planifica la procesión en la calle en función a la cantidad de Hermanos que participen. En esta pantalla, se podrá editar la sección a la que van y la insignia que portarán.

The screenshot displays the 'tunicAPP' web interface. At the top, there is a navigation bar with links: PERSONAS, CONTROL DE SALIDA, HERMANOS, and TÚNICAS. The main header features the 'tunicAPP' logo and the tagline 'Una herramienta que te ayudará a llevar el control de tus hermanos'. Below this, the title 'Editar Control de Salida' is centered. The form is divided into several sections: 1. 'Panel de Búsqueda' with a 'DNI/NIE (*)' field containing '30536271E' and 'Buscar'/'Limpiar' buttons. 2. 'Datos de Bea Rodriguez' showing fields for 'DNI/NIE' (30536271E), 'Nombre' (Bea), and 'Apellidos' (Rodriguez). 3. 'Control de Salida' with fields for 'Tipo de Recibo (*)' (Salida Profesional 2019), 'Cuantía' (20), and 'Pagado' (No), along with an 'Establecer como pagado' button. 4. 'Túnica' section with fields for 'Titular' (VIRGEN), 'Túnica' (V-3 de 123cm por 123cm), 'Portador' (a dropdown menu), and 'Sección'. At the bottom, there is an 'Editar Control de Salida' button, social media icons for Twitter, Facebook, and YouTube, and a copyright notice: 'Copyright 2019 - Jesús Manuel García Lora'.

Figura 4.13: Editar Control de Salida

■ Borrar Control de Salida

En este módulo el administrador del sistema ofrece una pantalla de búsqueda de la persona mediante un DNI facilitado. El sistema hará la búsqueda del DNI y borrará el control de salida asociado a nuestra base de datos mostrando un mensaje satisfactorio en caso positivo.



Figura 4.14: Borrar Control de Salida

4.4.3.4. Estadísticas

En el módulo de Control de Salida obtenemos un módulo de estadísticas. En dicho módulo, podemos tener de forma gráfica información obtenida tanto del histórico de la Hermandad como del año en curso.

Del mundo de la estadística todos coincidimos en que podemos sacar varias ramas de desarrollo y documentación, pero para este proyecto nos hemos limitado a obtener tres tipos de información gráficas:

- **Intervalos de Edad** En este tipo de gráfico, representado de forma circular, se obtiene una información basada en las edades de los Hermanos que participan en la salida procesional. Los intervalos, están definidos en:
 - De 0 a 18 años
 - De 19 a 30 años
 - De 31 a 50 años
 - Mayores de 50
- **Número de Controles** En este tipo de gráfico, representado en forma de líneas, se obtiene una información basada en el número de controles dados de alta tanto del año actual como del histórico desde 2015. Dicha información se obtiene de datos privados de la Hermandad.
- **Género de los Hermanos** En este tipo de gráfico, representado en forma de barras, se obtiene una información basada en la cantidad de hombres y mujeres que participan en nuestra procesión tanto del año actual como del histórico desde 2015. Dicha información, se obtiene de datos privados de la Hermandad.

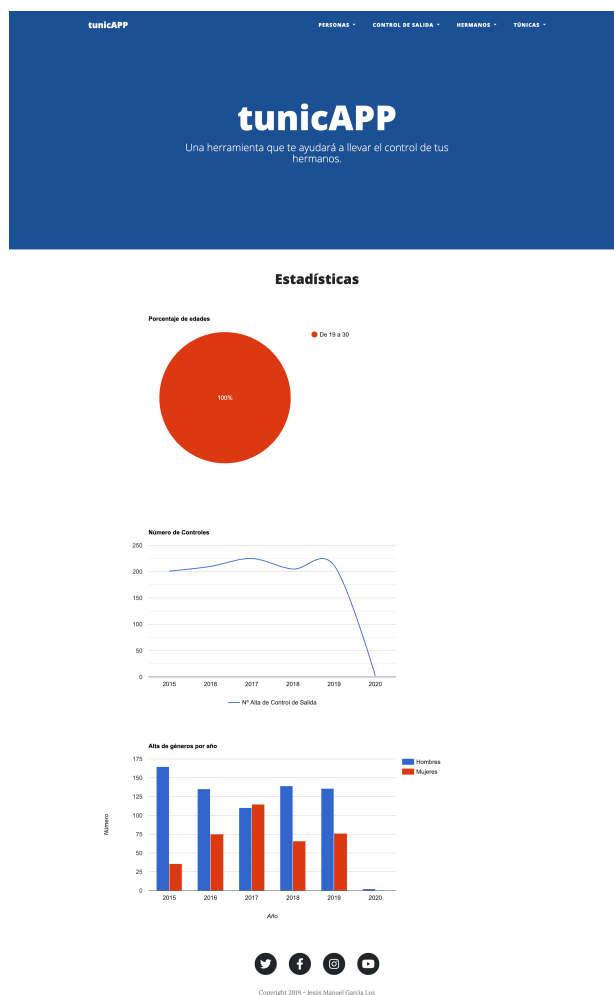


Figura 4.15: Estadísticas

4.4.3.5. Túnicas

En este módulo el administrador tendrá la labor de gestionar las túnicas pertenecientes a la Hermandad. En este caso, las túnicas son propiedad de la Hermandad y para facilitar la labor de adjudicación de las mismas se definen por tipo de paso (Cristo o Virgen, ya que son de diferente color), largo y ancho. Así, de un vistazo, el responsable de asignar túnicas puede ver las que se encuentran disponibles.

Los subniveles asociados al módulo de Túnicas son los siguientes:

- **Lista de Túnicas**

En este módulo obtendremos la lista total de túnicas dadas de alta en el sistema además de datos que se muestran de forma clara y concisa en una tabla para poder hacer uso de la información en el momento de consulta.

Por otro lado, la tabla ofrece la posibilidad de exportar los listados a Excel o a PDF facilitando así su exportación y trabajo fuera del sistema. Esta utilidad es muy útil por el administrador para suministrar a sus ayudantes de los listados pertinentes.

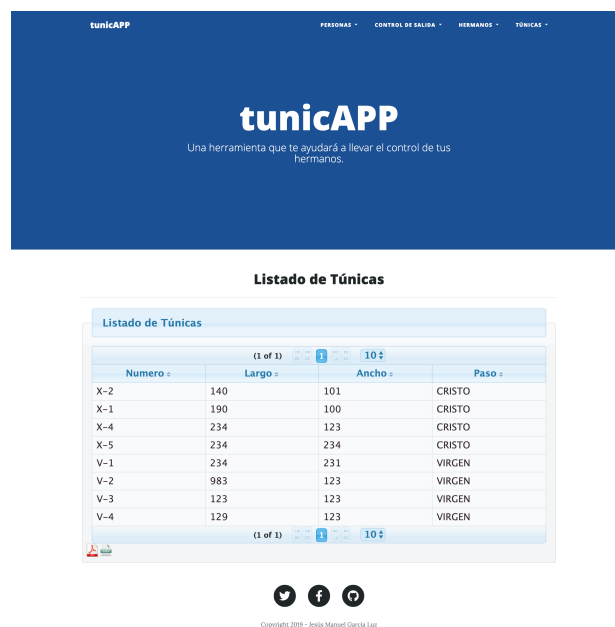


Figura 4.16: Listado de Túnicas.

■ Añadir Túnica

El sistema ofrece un formulario para poder dar de alta una túnica. En dicho formulario cabe destacar el campo Número, ya que se trata de una numeración interna con la que trabaja la cofradía para el control de la misma siendo su composición de X(Cristo) o V(Virgen) - número. Ejemplo: X-21. Esto le sirve a ellos para poder contabilizar de forma rápida la cantidad de túnicas que tienen de un paso o de otro. Con tunicAPP, se soluciona dicho problema.

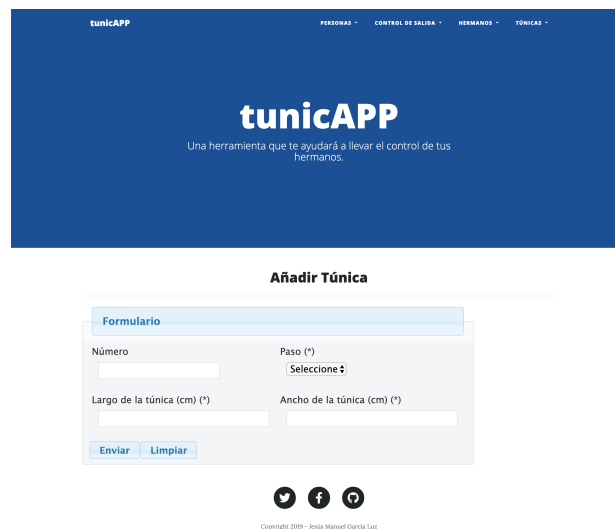


Figura 4.17: Añadir Túnica

■ Editar Túnica

El sistema mostrará una búsqueda en función al número interno que la cofradía usa para hacer referencia a sus túnicas explicado en el módulo .Añadir Túnica”. Se decidió hacer uso del número interno para hacer la aplicación mas amena y cercana a las personas que la utilicen.

tunicAPP PERSONAS CONTROL DE SALIDA HERMANOS TÚNICAS

tunicAPP
Una herramienta que te ayudará a llevar el control de tus hermanos.

Editar Túnica

Panel de Búsqueda

Número
X-2
Buscar Limpiar

Datos de la túnica X-2:

Número X-2	Paso CRISTO
Largo de la túnica (cm) 140	Ancho de la túnica (cm) 101

Editar

Twitter Facebook YouTube
Copyright 2019 - Jesús Manuel García Lora

Figura 4.18: Editar Túnica

- **Borrar Túnica** En este módulo el responsable de la aplicación podrá introducir el número interno de túnica a borrar, eliminándola del sistema.

tunicAPP PERSONAS CONTROL DE SALIDA HERMANOS TÚNICAS

tunicAPP
Una herramienta que te ayudará a llevar el control de tus hermanos.

Eliminar Túnica

Panel de Búsqueda

Número
Borrar Limpiar

Twitter Facebook YouTube
Copyright 2019 - Jesús Manuel García Lora

Figura 4.19: Borrar Túnica

4.5. Tipos de usuarios

Se pueden encontrar distintos roles dentro de aplicación, donde cada uno tendrá unos permisos para los diferentes módulos dentro de la herramienta. Son los siguientes:

- **Administrador:**

Es el perfil encargado de gestionar la Hermandad. Se encarga de tareas laboriosas como el alta de Personas y Hermanos, Controles de Salida y Túnicas. Además, es el responsable de registrar en el sistema los datos por lo que está comprometido en ofrecer una información al sistema clara, concisa y real.

- **Hermano:**

Es el perfil del hermano perteneciente a la cofradía. En esta versión, podemos encontrar la posibilidad de consultar y pagar tus cuotas, además de poder reservar nuestro control de salida.

4.6. Interfaz de usuario

Como se ha comentado en el apartado anterior, la aplicación posee una vista enfocada hacia los diferentes tipos de usuarios que posee la herramienta, haciendo accesibles los módulos y acciones en estas en función del perfil seleccionado o el usuario introducido al usar la aplicación.

A continuación se indicarán los módulos que posee acceso cada usuario dentro de la herramienta y las acciones que puede realizar dentro de las mismas.

4.6.1. Administrador

La vista para el administrador incluye todos los módulos de la herramienta y posee permisos para hacer uso de la herramienta al completo, los módulos con sus acciones son los siguientes:

- Inicio
- Panel de Administrador
 - Hermanos
 - Listar los Hermanos
 - ◇ Exportar en Excel
 - ◇ Exportar en PDF
 - Alta de Hermano
 - Baja de Hermano
 - Edición del Hermano
 - Baja del Hermano
 - Personas
 - Listar las Personas
 - ◇ Exportar en Excel
 - ◇ Exportar en PDF
 - Alta de la Persona
 - Baja de la Persona
 - Edición de la Persona
 - Baja de la Persona

- Control de Salida
 - Listar los Controles de Salida
 - ◊ Exportar en Excel
 - ◊ Exportar en PDF
 - Alta del Control de Salida
 - Edición del Control de Salida
 - Baja del Control de Salida
 - Estadísticas
- Túnicas
 - Listar las túnicas disponibles
 - ◊ Exportar en Excel
 - ◊ Exportar en PDF
 - Alta de túnica
 - Edición de túnica
 - Eliminar una túnica

4.6.2. Usuario

La vista para el usuario incluye todos los módulos de la herramienta y pero posee permisos reducidos en algunos de los módulos de la misma, los módulos con sus acciones son los siguientes:

- Inicio
- Panel del Hermano
 - Tus Cuotas
 - Reserva tu Control de Salida

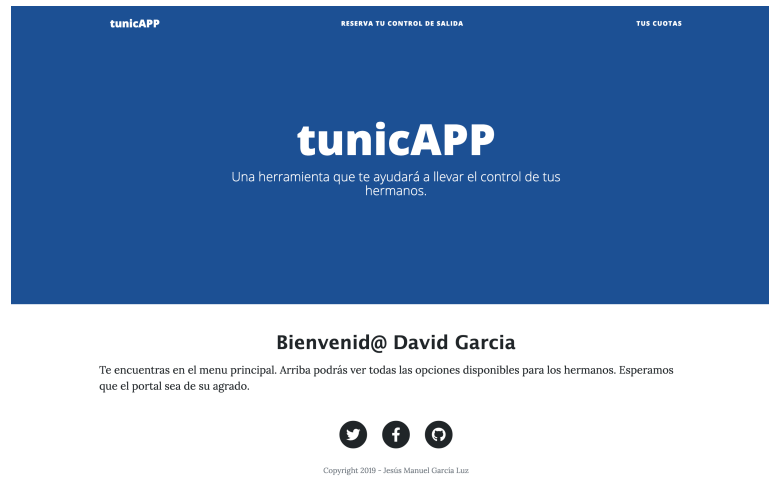


Figura 4.20: Panel de Hermano

4.7. Lenguajes de programación y tecnologías

En este apartado se desarrollan los diferentes lenguajes y tecnologías usados para la creación del presente TFG, incluyendo el documento del proyecto, así como sus anexos y el porqué de su uso en el mismo.

4.7.1. Java

Lenguaje principal para el desarrollo de la herramienta, el cual posee propiedades beneficiosas para el desarrollo web que fueron desglosadas en el capítulo 2 de este documento.



Figura 4.21: Logo Java

4.7.2. Ajax

Ajax es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas.



Figura 4.22: Logo Ajax.

La herramienta crea barras de progreso en vista del usuario para indicar el progreso de subida de archivos hacia la aplicación, indicándole al usuario cuando ha terminado dicho proceso. Además realiza el filtrado de ciertos selectores en la herramienta.

4.7.3. SQL

SQL es un lenguaje específico de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos. Este lenguaje permite efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas. Nosotros usaremos una adaptación llamada **mySQL**.



Figura 4.23: Logo mySQL.

La herramienta hace consultas hacia bases de datos externas para recuperar información relevante para con la misma y poder mejorar la eficiencia de la aplicación y otorgarle un uso más sencillo auto-completando información en sus formularios.

4.7.4. Latex

Latex es un sistema de composición de textos, orientado a la creación de documentos escritos, el cual es usado de forma especialmente intensa en la generación de artículos y libros científicos que incluyen, entre otros elementos, expresiones matemáticas.



Figura 4.24: Logo Latex.

Latex es el sistema elegido para la generación de la documentación de este proyecto en su totalidad.

4.7.5. Selenium

Selenium es un framework usado para el desarrollo de pruebas automatizadas que puede escribirse en lenguaje Java. Se incluye como librería dentro del proyecto y vamos a usarlo para realizar una batería de pruebas de regresión para nuestro desarrollo ofreciéndole un gran valor al producto aportado.



Figura 4.25: Logo Latex.

4.8. Herramientas

En este apartado se enumeran las herramientas utilizadas en la creación del presente proyecto, incluyendo el documento del mismo, así como sus anexos y aplicaciones secundarias para comprobar el buen funcionamiento de este proyecto.

4.8.1. JSF

JSF es un marco de trabajo para crear aplicaciones java J2EE basadas en el patron MVC de tipo 1. JSF tiene como características principales:

- Utiliza páginas JSP para generar las vistas, añadiendo una biblioteca de etiquetas propia para crear los elementos de los formularios
- Asocia a cada vista con formularios un conjunto de objetos java manejados por el controlador (managed beans) que facilitan la recogida, manipulación y visualización de los valores mostrados en los diferentes elementos de los formularios.
- Introduce una serie de etapas en el procesamiento de la petición, como por ejemplo la de validación, reconstrucción de la vista, recuperación de los valores de los elementos, etc.
- Utiliza un sencillo fichero de configuración para el controlador en formato xml
- Es extensible, pudiendo crearse nuevos elementos de la interfaz o modificar los ya existentes.
- Y lo que es más importante: forma parte del estándar J2EE. En efecto, hay muchas alternativas para crear la capa de presentación y control de una aplicación web java, como Struts y otros frameworks, pero solo JSP forma parte del estándar.

4.8.1.1. Porqué utilizar JSF

JSF nos permite desarrollar rápidamente aplicaciones de negocio dinámicas en las que toda la lógica de negocio se implementa en Java. JSF nos ofrece una serie de ventajas:

- JSF forma parte del estándar J2EE, mientras que otras tecnologías para creación de vistas de las aplicaciones no lo forman, como por ejemplo Struts.
- JSF dispone de varias implementaciones diferentes, incluyendo un conjunto de etiquetas y APIs estándar que forman el núcleo del framework
- Lenguaje OpenSource apoyado por una gran comunidad



Figura 4.26: Logo JSF.

4.8.2. JPA

JPA es la propuesta estándar que ofrece Java para implementar un Framework Object Relational Mapping (ORM), que permite interactuar con la base de datos por medio de objetos, de esta forma, JPA es el encargado de convertir los objetos Java en instrucciones para el Manejador de Base de Datos (MDB).

4.8.3. Primefaces

PrimeFaces es una biblioteca de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Como principales propiedades tenemos:



Figura 4.27: Logo JPA.

- Soporte nativo de Ajax, incluyendo Push/Comet.
- Kit para crear aplicaciones web para móviles.
- Es compatible con otras librerías de componentes, como JBoss
- Es un proyecto open source, activo y bastante estable entre versiones. RichFaces.



Figura 4.28: Logo PrimeFaces.

4.8.4. Maven

Apache Maven es una herramienta que estandariza la configuración de un proyecto en todo su ciclo de vida, como por ejemplo en todas las fases de compilación y empaquetado y la instalación de mecanismos de distribución de librerías, para que puedan ser utilizadas por otros desarrolladores y equipos de desarrollo. Principales características:

- Es software libre, con lo cual es el código está disponible, se podría modificar y customizar en caso de que fuera necesario.
- Fomenta la reutilización de código y de librerías. El hecho de que Apache Maven ofrezca repositorios oficiales y públicos de software libre, con librerías desplegadas, que toda la comunidad de desarrolladores de software utiliza, hace que este concepto también pueda trasladarse al mundo empresarial, a través de repositorios remotos corporativos, compartidos por distintos equipos de proyectos o el propio equipo de desarrollo.
- Es multi-plataforma, puede funcionar tanto en entornos Linux como Windows al ser una aplicación Java.



Figura 4.29: Logo Maven.

4.8.5. Tomcat Apache

Tomcat es un servidor Web con soporte para servlets y JSPs. No es un servidor de aplicaciones, como JBoss o JOnAS. Trae incluido el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor Web Apache

4.8.6. GitHub

Es un repositorio Git para almacenar de forma segura nuestro código. Para este proyecto, se ha creado el siguiente repositorio donde puedes encontrar todo el código fuente. <https://github.com/YiseKai/TFG/>



Figura 4.30: Logo TomCat Apache.



Figura 4.31: Logo GitHub.

Capítulo 5

Análisis

En este capítulo se realizará un estudio de la herramienta que se desea desarrollar, comenzado desde la concepción del proyecto, las necesidades y prioridades que precisa la aplicación y profundizando en la navegabilidad que debe tener dicha herramienta para cumplir con los objetivos y necesidades que se esperan de este proyecto.

5.1. Requisitos

En este apartado se indican los requisitos para la herramienta, los cuales servirán para comprobar cuales son las necesidades que precisa la misma y los problemas que pretende solventar. Este estudio sobre la herramienta, servirá a su vez como comprobante sobre los objetivos fijados a la hora de la creación del proyecto, siendo necesario la re-evaluación de estos objetivos al finalizar esta etapa del desarrollo de la aplicación, para constatar que cumple lo acordado inicialmente.

5.1.1. Requisitos funcionales

Los requisitos funcionales son aquellos requisitos que indican una funcionalidad dentro de un software, es decir, explican el comportamiento de la herramienta frente a una acción.

Para esta herramienta, los requisitos funcionales se expresan en forma de acciones que el usuario debe poder realizar con la herramienta, donde esta realizará ciertos procesos y devolverá información al usuario de una forma u otra según sea el caso.

Los requisitos funcionales para esta herramienta son los siguientes:

- **Inicio de sesión en la herramienta**

La herramienta debe ser capaz de diferenciar entre varios tipos de usuarios para que estos realicen acciones dentro de la herramienta, según sea su rol en la aplicación. Estos perfiles serán: “Administrador”, “Hermano” ’.

- **Consultar Listado de Personas**

La herramienta debe ser capaz de mostrar una consulta sobre el listado de personas existentes en el sistema.

- **Añadir una Persona**

La herramienta debe ser capaz de mostrar un formulario para poder añadir una Persona al sistema.

- **Editar una Persona**

La herramienta debe ser capaz de mostrar un formulario con los datos de una persona pre-cargada para su edición.

- **Borrar una Persona**
La herramienta debe ser capaz de borrar una persona facilitada en el sistema.
- **Consultar Listado de Hermanos**
La herramienta debe ser capaz de mostrar una consulta sobre el listado de hermanos existentes en el sistema.
- **Añadir un Hermano**
La herramienta debe ser capaz de mostrar un formulario para poder añadir un hermano al sistema.
- **Editar un Hermano**
La herramienta debe ser capaz de mostrar un formulario con los datos del hermano precargados para su edición.
- **Borrar un Hermano**
La herramienta debe ser capaz de borrar un hermano facilitado en el sistema.
- **Consultar Listado de Controles de Salidas**
La herramienta debe ser capaz de mostrar una consulta sobre el listado de controles de salidas existentes, tanto de Cristo como de Virgen, y la cantidad total de controles de salida dados de alta en el sistema.
- **Añadir Control de Salida**
La herramienta debe ser capaz de mostrar un formulario para poder añadir un control de salida al sistema.
- **Editar un Control de Salida**
La herramienta debe ser capaz de mostrar un formulario con los datos precargados del control de salida para su edición.
- **Borrar un Control de Salida**
La herramienta debe ser capaz de borrar un control de salida facilitado en el sistema.
- **Estadísticas**
La herramienta debe ser capaz de generar gráficas asociadas a la información depositada en la BD.
- **Consultar Listado de Túnica**
La herramienta debe ser capaz de mostrar una consulta sobre el listado de túnicas existentes en el sistema.
- **Añadir Túnica**
La herramienta debe ser capaz de mostrar un formulario para poder añadir una túnica en el sistema.
- **Editar Túnica**
La herramienta debe ser capaz de mostrar un formulario con los datos precargados para su edición.
- **Borrar Túnica**
La herramienta debe ser capaz de borrar una túnica facilitada en el sistema.
- **Marcar recibos como pagado**
La herramienta debe ser capaz de editar recibos tanto de hermano como de control de salida pudiendolos marcar como pagados.
- **Asignar ocupación**
La herramienta debe ser capaz de asignar una ocupación en el control de salida de un hermano.
- **Asignar sección**
La herramienta debe ser capaz de asignar una sección en el control de salida de un hermano.

- **Asignar Túnica**
La herramienta debe ser capaz de asignar una túnica a un hermano para su control de salida.
- **Exportar Listados en PDF**
La herramienta debe ser capaz de exportar en PDF la consulta de listados que ofrece la herramienta.
- **Exportar Listados en Excel**
La herramienta debe ser capaz de exportar en Excel la consulta de listados que ofrece la herramienta.
- **Mostrar las cuotas**
La herramienta debe ser capaz de mostrar las cuotas pertenecientes al hermano que se ha logueado en el sistema.
- **Poder reservar el control de salida**
La herramienta debe ser capaz de poder reservar el control de salida al hermano que se ha logueado en el sistema.
- **Pago mediante PayPal**
La herramienta debe ser capaz de realizar un pago de recibo mediante PayPal en las opciones disponibles al rol Hermano.
- **Introducción de nuevos registros en la herramienta**
La herramienta debe ser capaz de introducir nuevos registros en el sistema para el manejo de la información.
- **Modificación de registros en la herramienta**
La herramienta debe ser capaz de modificar los registros del sistema para el manejo de la información.
- **Eliminación de registros en la herramienta**
La herramienta debe ser capaz de borrar los registros del sistema para el manejo de la información.
- **Consultas**
La herramienta debe ser capaz de realizar consultas sobre los datos introducidos en la aplicación de manera sencilla y rápida.

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Dentro del sistema se encuentran los siguientes subsistemas:

5.1.2. Requisitos de información

Los requisitos de información son aquellos que indican los datos que posee y maneja un software, es decir, el contenido de la información de la herramienta y las acciones que se pueden realizar sobre ellos.

Los requisitos de información para esta herramienta son los siguientes:

- La herramienta debe crear una BD donde será almacenada la información para ser tratada por el sistema.
- La herramienta debe ser capaz de consultar los datos de la BD creada para ser tratada por el sistema.
- La herramienta debe ser capaz de entregar listados en Excel y en PDF.
- La herramienta debe permitir una comunicación con PayPal
- La herramienta debe ser capaz de generar gráficas para representar la información de los controles de salida.

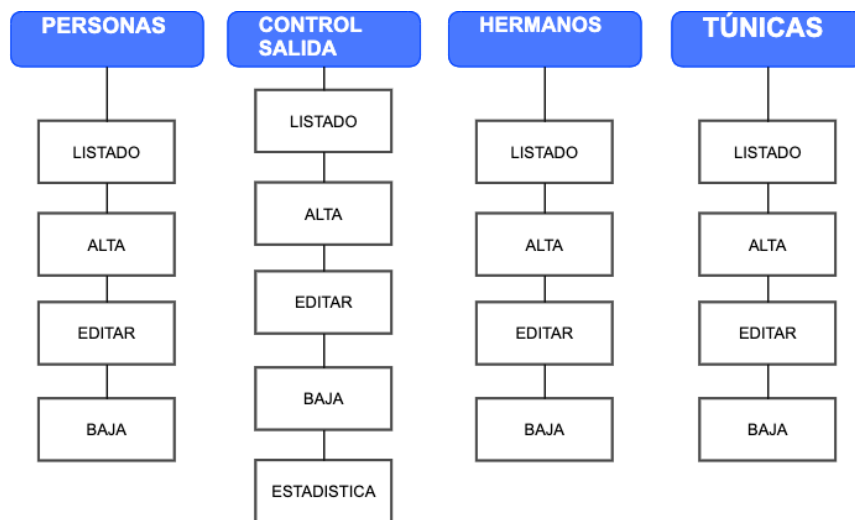


Figura 5.1: Diagrama para Administrador.

- La herramienta debe permitir la comunicación con una pasarela de pago.

5.1.3. Requisitos no funcionales

Los requisitos no funcionales son aquellos que indican las características básicas necesarias en un software, es decir, propiedades impuestas para la respuesta óptima de un software que ha sido desarrollado bajo demanda.

Los requisitos no funcionales para esta herramienta son los siguientes:

- **Usabilidad**
La herramienta debe poseer una visualización sencilla que optimice el uso de la misma, así como facilitar un manual para el uso a nivel de usuario.
- **Disponibilidad**
La herramienta debe ser accesible de forma ininterrumpida, sin importar la hora ni el día para su uso.
- **Rendimiento**
La herramienta debe funcionar de la forma más eficiente posible, incluso cuando el flujo de datos sobre la misma sea alto.
- **Tiempo de respuesta**
La herramienta debe ser capaz de reaccionar a su uso en el menor tiempo posible, intentando evitar en su manejo, cualquier deficiencia temporal que acarren la administración de sus datos.
- **Seguridad**
La herramienta debe proporcionar integridad y seguridad a los datos introducidos en la misma, así como, los recibidos por medio de consultas externas. Indicar que, los datos que procesa la herramienta son confidenciales y privados.
- **Mantenibilidad**
La herramienta debe ser fácil de mantener para los usuarios desarrolladores o los técnicos para que el funcionamiento de la herramienta sea siempre óptimo.

5.2. Análisis del sistema

En este apartado del presente documento, se desarrollarán los diferentes casos de uso que posee la herramienta, así como, los diagramas de secuencia para la realización de las acciones dentro de la aplicación.

5.2.1. Casos de uso

Los casos de uso (CU) describen los pasos o acciones que deben de realizarse para llevar a cabo un proceso. Los casos de uso poseen elementos que interactúan con los sistemas llamados actores, los cuales pueden ser, usuarios de la herramienta, entidades o sistemas externos (como bases de datos) que participan en el uso de la herramienta.

Para este proyecto y, como la herramienta posee dos perfiles de usuarios diferentes, encontraremos 2 actores en la aplicación y numerosos casos de uso para la misma. Las siguientes figuras indican los actores y su relación con la herramienta.

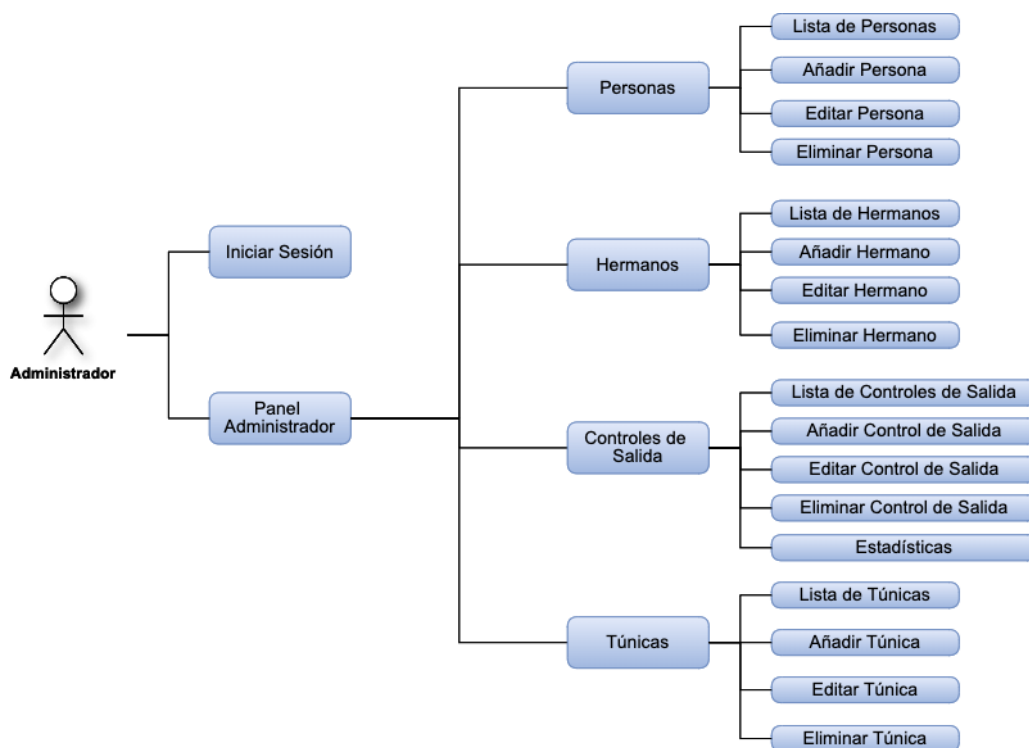


Figura 5.2: Administrador: Diagrama de casos de uso.

A continuación se desglosarán los diferentes casos de uso enfocados a como deben ser en la he-

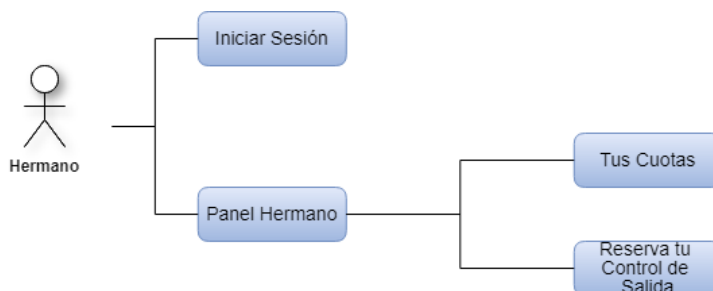


Figura 5.3: Hermano: Diagrama de casos de uso.

herramienta haciendo uso para ello, de las acciones vistas en el capítulo anterior y, enmarcándolas dentro de los casos de uso indicados en las figuras anteriores, las cuales muestran los diferentes casos de uso para cada perfil de usuario en la aplicación.

5.2.2. Casos de uso - Entrar

<i>CU-001</i>	<i>Entrar</i>
<i>Actores</i>	Administrador, Hermano.
<i>Tipo de CU</i>	Entrar
<i>Precondiciones</i>	No aplica
<i>Postcondiciones</i>	Carga de pantalla asociada al Inicio de Sesión
<i>Escenario</i>	<ul style="list-style-type: none"> - El usuario hace click en el botón Entrar - La herramienta realiza una navegación a la pantalla de inicio de sesión
<i>Escenario alternativo</i>	No aplica

Tabla 5.1: CU-001: Entrar

5.2.3. Casos de uso - Iniciar Sesión

<i>CU-002</i>	<i>Iniciar Sesión</i>
<i>Actores</i>	Administrador, Hermano.
<i>Tipo de CU</i>	Iniciar Sesión
<i>Precondiciones</i>	El usuario debe estar registrado en la herramienta.
<i>Postcondiciones</i>	El usuario inicia sesión en la herramienta.
<i>Escenario</i>	<ul style="list-style-type: none"> - El usuario entra en el módulo de Inicio de Sesión de la herramienta. - La herramienta muestra la pantalla de Inicio de Sesión. - El usuario introduce sus credenciales en el formulario - La herramienta redirige al panel que le pertenezca a su rol
<i>Escenario alternativo</i>	El usuario introdujo valores inválidos en el formulario de inicio de sesión. La herramienta indicará los errores.

Tabla 5.2: CU-002: Inicio de Sesión

5.2.4. Casos de uso - Personas

CU-003	Personas
Actores	Administrador.
Tipo de CU	Panel de Administrador
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de listado de Personas.
Escenario	<ul style="list-style-type: none"> - El usuario hace click al menú Personas - La herramienta realiza la navegación a la pantalla de Listado de Personas.
Escenario alternativo	No aplico

Tabla 5.3: CU-003: Personas

5.2.5. Casos de uso - Control de Salida

CU-004	Control de Salida
Actores	Administrador.
Tipo de CU	Panel de Administrador
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de listado de Control de Salida.
Escenario	<ul style="list-style-type: none"> - El usuario hace click al menú Control de Salida - La herramienta realiza la navegación a la pantalla de Listado de Control de Salida.
Escenario alternativo	No aplico

Tabla 5.4: CU-004: Control de Salida

5.2.6. Casos de uso - Hermanos

CU-005	Hermanos
Actores	Administrador.
Tipo de CU	Panel de Administrador
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de listado de Hermanos.
Escenario	<ul style="list-style-type: none"> - El usuario hace click al menú Hermanos - La herramienta realiza la navegación a la pantalla de Listado de Hermanos.
Escenario alternativo	No aplico

Tabla 5.5: CU-005: Hermanos

5.2.7. Casos de uso - Túnicas

CU-006	<i>Túnicas</i>
Actores	Administrador.
Tipo de CU	Panel de Administrador
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol <u>Administrador</u> .
Postcondiciones	La herramienta hace navegación hasta la pantalla de listado de Túnicas.
Escenario	<ul style="list-style-type: none"> - El usuario hace click al menú Túnicas - La herramienta realiza la navegación a la pantalla de Listado de Túnicas.
Escenario alternativo	No aplico

Tabla 5.6: CU-006: Túnicas

5.2.8. Casos de uso - Reserva tu Control de Salida

CU-007	<i>Reserva tu Control de Salida</i>
Actores	Hermano.
Tipo de CU	Panel de Hermano
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol <u>Hermano</u> .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Reserva tu Control de Salida.
Escenario	<ul style="list-style-type: none"> - El usuario hace click al menú Reserva tu Control de Salida - La herramienta realiza la navegación a la pantalla de Reserva tu Control de Salida - El usuario cumplimenta el formulario y da de alta la reserva del Control de Salida
Escenario alternativo	No aplico

Tabla 5.7: CU-007: Reserva tu Control de Salida

CU-008	<i>Reserva tu Control de Salida</i>
Actores	Hermano.
Tipo de CU	Panel de Hermano
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Hermano .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Reserva tu Control de Salida.
Escenario	<ul style="list-style-type: none"> - El usuario hace click al menú Reserva tu Control de Salida - La herramienta realiza la navegación a la pantalla de Reserva tu Control de Salida - El usuario cumplimenta el formulario y da de alta la reserva del Control de Salida
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.8: CU-008: Reserva tu Control de Salida

5.2.9. Casos de uso - Tus cuotas

CU-009	<i>Tus cuotas</i>
Actores	Hermano.
Tipo de CU	Panel de Hermano
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Hermano .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Tus Cuotas.
Escenario	<ul style="list-style-type: none"> - El usuario hace click al menú Tus Cuotas - La herramienta realiza la navegación a la pantalla de Tus Cuotas - El usuario cumplimenta el formulario
Escenario alternativo	No aplica

Tabla 5.9: CU-009: Tus Cuotas

CU-010	<i>Tus cuotas</i>
Actores	Hermano.
Tipo de CU	Panel de Hermano
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Hermano .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Tus Cuotas.
Escenario	<ul style="list-style-type: none"> - El usuario hace click al menú Tus Cuotas - La herramienta realiza la navegación a la pantalla de Tus Cuotas - El usuario cumplimenta el formulario
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.10: CU-010: Tus Cuotas

5.2.10. Casos de uso - Listado de Personas

CU-011	Listado de Personas
Actores	Administrador.
Tipo de CU	Personas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Listado de Personas.
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Personas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Personas - La herramienta gestiona el evento y realiza la navegabilidad
Escenario alternativo	No aplica

Tabla 5.11: CU-011: Listado de Personas

5.2.11. Casos de uso - Exportar a PDF el Listado de Personas

CU-012	Exportar a PDF el Listado de Personas
Actores	Administrador.
Tipo de CU	Personas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador .
Postcondiciones	La herramienta exporta un documento PDF con la información del Listado
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Personas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Personas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario hace click en el icono de PDF situado en el listado - La herramienta recoge la información del listado y exporta un PDF para el usuario
Escenario alternativo	No aplica

Tabla 5.12: CU-012: Exportar a PDF el Listado de Personas

5.2.12. Casos de uso - Exportar a Excel el Listado de Personas

CU-013	<i>Exportar a Excel el Listado de Personas</i>
Actores	Administrador.
Tipo de CU	Personas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol <u>Administrador</u> .
Postcondiciones	La herramienta exporta un documento Excel con la información del <u>Listado</u>
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Personas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Personas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario hace click en el icono de Excel situado en el listado - La herramienta recoge la información del listado y exporta un Excel para el usuario
Escenario alternativo	No aplico

Tabla 5.13: CU-013: Exportar a Excel el Listado de Personas

5.2.13. Casos de uso - Añadir Persona

CU-014	<i>Añadir Persona</i>
Actores	Administrador.
Tipo de CU	Personas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol <u>Administrador</u> .
Postcondiciones	La herramienta hace navegación hasta la pantalla de <u>Añadir Persona</u>
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Personas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Añadir Persona - La herramienta gestiona el evento y realiza la navegabilidad - El usuario cumplimenta el formulario
Escenario alternativo	No aplico

Tabla 5.14: CU-014: Añadir Persona

CU-015	<i>Añadir Persona</i>
Actores	Administrador.
Tipo de CU	Personas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol <u>Administrador</u> .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Añadir Persona
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Personas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Añadir Persona - La herramienta gestiona el evento y realiza la navegabilidad - El usuario cumplimenta el formulario
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.15: CU-015: Añadir Persona

5.2.14. Casos de uso - Editar Persona

CU-016	<i>Editar Persona</i>
Actores	Administrador.
Tipo de CU	Personas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol <u>Administrador</u> .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Persona
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Personas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Personas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a editar - La herramienta muestra los datos asociados a dicho DNI - El usuario edita los campos que necesita editar
Escenario alternativo	No aplica

Tabla 5.16: CU-016: Editar Persona

CU-017	<i>Editar Persona</i>
Actores	Administrador.
Tipo de CU	Personas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol <u>Administrador</u> .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Persona
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Personas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Personas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a editar - La herramienta muestra los datos asociados a dicho DNI - El usuario edita los campos que necesita editar
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta <u>detalla los motivos de los errores en el mismo.</u>

Tabla 5.17: CU-017: Editar Persona

CU-018	<i>Editar Persona</i>
Actores	Administrador.
Tipo de CU	Personas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol <u>Administrador</u> .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Persona
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Personas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Personas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a editar - La herramienta muestra los datos asociados a dicho DNI - El usuario edita los campos que necesita editar
Escenario alternativo	El sistema no encuentra en la BD el DNI facilitado por el usuario.

Tabla 5.18: CU-018: Editar Persona

5.2.15. Casos de uso - Eliminar Persona

CU-019	<i>Eliminar Persona</i>
Actores	Administrador.
Tipo de CU	Personas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Persona
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Personas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Personas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a Eliminar y hace click en Eliminar - La herramienta elimina los registros asociados en cascada de dicho DNI
Escenario alternativo	No aplica

Tabla 5.19: CU-019: Eliminar Persona

CU-020	<i>Eliminar Persona</i>
Actores	Administrador.
Tipo de CU	Personas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Persona
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Personas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Personas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a Eliminar y hace click en Eliminar - La herramienta elimina los registros asociados en cascada de dicho DNI
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.20: CU-020: Eliminar Persona

CU-021	<i>Eliminar Persona</i>
Actores	Administrador.
Tipo de CU	Personas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Persona
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Personas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Personas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a Eliminar y hace click en Eliminar - La herramienta elimina los registros asociados en cascada de dicho DNI
Escenario alternativo	El sistema no encuentra en la BD el DNI facilitado por el usuario.

Tabla 5.21: CU-021: Eliminar Persona

5.2.16. Casos de uso - Listado de Túnicas

CU-022	<i>Listado de Túnicas</i>
Actores	Administrador.
Tipo de CU	Túnicas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Listado de Túnicas.
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Túnicas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Túnicas - La herramienta gestiona el evento y realiza la navegabilidad
Escenario alternativo	No aplico

Tabla 5.22: CU-022: Listado de Túnicas

5.2.17. Casos de uso - Exportar a PDF el Listado de Túnicas

CU-023	<i>Exportar a PDF el Listado de Túnicas</i>
Actores	Administrador.
Tipo de CU	Túnicas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta exporta un documento PDF con la información del Listado
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Túnicas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Túnicas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario hace click en el icono de PDF situado en el listado - La herramienta recoge la información del listado y exporta un PDF para el usuario
Escenario alternativo	No aplico

Tabla 5.23: CU-023: Exportar a PDF el Listado de Túnicas

5.2.18. Casos de uso - Exportar a Excel el Listado de Túnicas

CU-024	<i>Exportar a Excel el Listado de Túnicas</i>
Actores	Administrador.
Tipo de CU	Túnicas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta exporta un documento Excel con la información del Listado
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Túnicas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Túnicas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario hace click en el icono de Excel situado en el listado - La herramienta recoge la información del listado y exporta un Excel para el usuario
Escenario alternativo	No aplico

Tabla 5.24: CU-024: Exportar a Excel el Listado de Túnicas

5.2.19. Casos de uso - Añadir Túnica

CU-025	Añadir Túnica
Actores	Administrador.
Tipo de CU	Túnicas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Añadir Túnica
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Túnicas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Añadir Túnica - La herramienta gestiona el evento y realiza la navegabilidad - El usuario cumplimenta el formulario
Escenario alternativo	No aplico

Tabla 5.25: CU-025: Añadir Túnica

CU-026	Añadir Túnica
Actores	Administrador.
Tipo de CU	Túnicas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Añadir Túnica
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Túnicas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Añadir Túnica - La herramienta gestiona el evento y realiza la navegabilidad - El usuario cumplimenta el formulario
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.26: CU-026: Añadir Túnica

5.2.20. Casos de uso - Editar Túnica

CU-027	<i>Editar Túnica</i>
Actores	Administrador.
Tipo de CU	Túnicas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Túnica
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Túnicas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Túnicas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el número de túnica a editar - La herramienta muestra los datos asociados a dicho número de túnica - El usuario edita los campos que necesita editar
Escenario alternativo	No aplica

Tabla 5.27: CU-027: Editar Túnica

CU-028	<i>Editar Túnica</i>
Actores	Administrador.
Tipo de CU	Túnicas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Túnica
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Túnicas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Túnicas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el número de túnica a editar - La herramienta muestra los datos asociados a dicho número de túnica - El usuario edita los campos que necesita editar
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.28: CU-028: Editar Túnica

CU-029	<i>Editar Túnica</i>
Actores	Administrador.
Tipo de CU	Túnicas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Túnica
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Túnicas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Túnicas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el número de túnica a editar - La herramienta muestra los datos asociados a dicho número de túnica - El usuario edita los campos que necesita editar
Escenario alternativo	El sistema no encuentra en la BD el número de túnica facilitado por el usuario.

Tabla 5.29: CU-029: Editar Túnica

5.2.21. Casos de uso - Eliminar Túnica

CU-030	<i>Eliminar Túnica</i>
Actores	Administrador.
Tipo de CU	Túnicas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Túnica
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Túnicas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Túnicas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el número de túnica a Eliminar - La herramienta elimina en cascada los datos asociados a dicha túnica
Escenario alternativo	No aplica

Tabla 5.30: CU-030: Eliminar Túnica

CU-031	<i>Eliminar Túnica</i>
Actores	Administrador.
Tipo de CU	Túnicas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol <u>Administrador</u> .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Túnica
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Túnicas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Túnicas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el número de túnica a Eliminar - La herramienta elimina en cascada los datos asociados a dicha túnica
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.31: CU-031: Eliminar Túnica

CU-032	<i>Eliminar Túnica</i>
Actores	Administrador.
Tipo de CU	Túnicas
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol <u>Administrador</u> .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Túnica
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Túnicas - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Túnicas - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el número de túnica a Eliminar - La herramienta elimina en cascada los datos asociados a dicha túnica
Escenario alternativo	El sistema no encuentra en la BD el número de túnica facilitado por el usuario.

Tabla 5.32: CU-032: Eliminar Túnica

5.2.22. CU-033: Listado de Hermanos

CU-033	Listado de Hermanos
Actores	Administrador.
Tipo de CU	Hermanos
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Listado de Hermanos.
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Hermanos - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Hermanos - La herramienta gestiona el evento y realiza la navegabilidad
Escenario alternativo	No aplica

Tabla 5.33: CU-033: Listado de Hermanos

5.2.23. Casos de uso - Exportar a PDF el Listado de Hermanos

CU-034	Exportar a PDF el Listado de Hermanos
Actores	Administrador.
Tipo de CU	Hermanos
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador .
Postcondiciones	La herramienta exporta un documento PDF con la información del Listado
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Hermanos - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Hermanos - La herramienta gestiona el evento y realiza la navegabilidad - El usuario hace click en el icono de PDF situado en el listado - La herramienta recoge la información del listado y exporta un PDF para el usuario
Escenario alternativo	No aplica

Tabla 5.34: CU-034: Exportar a PDF el Listado de Hermanos

5.2.24. Casos de uso - Exportar a Excel el Listado de Hermanos

CU-035	<i>Exportar a Excel el Listado de Hermanos</i>
Actores	Administrador.
Tipo de CU	Hermanos
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta exporta un documento Excel con la información del Listado
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Hermanos - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Hermanos - La herramienta gestiona el evento y realiza la navegabilidad - El usuario hace click en el icono de Excel situado en el listado - La herramienta recoge la información del listado y exporta un Excel para el usuario
Escenario alternativo	No aplica

Tabla 5.35: CU-035: Exportar a Excel el Listado de Hermanos

5.2.25. Casos de uso - Añadir Hermano

CU-036	<i>Añadir Hermano</i>
Actores	Administrador.
Tipo de CU	Hermanos
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El hermano a dar de alta debe existir en el sistema como Persona.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Añadir Hermano
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Hermanos - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Añadir Hermano - La herramienta gestiona el evento y realiza la navegabilidad - El usuario cumplimenta el formulario
Escenario alternativo	No aplica

Tabla 5.36: CU-036: Añadir Hermano

CU-037	<i>Añadir Hermano</i>
Actores	Administrador.
Tipo de CU	Hermanos
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El hermano a dar de alta debe existir en el sistema como Persona.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Añadir Hermano
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Hermanos - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Añadir Hermano - La herramienta gestiona el evento y realiza la navegabilidad - El usuario cumplimenta el formulario
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.37: CU-037: Añadir Hermano

5.2.26. Casos de uso - Editar Hermano

CU-038	<i>Editar Hermano</i>
Actores	Administrador.
Tipo de CU	Hermanos
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El hermano a editar debe formar parte del sistema como Persona.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Hermano
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Hermanos - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Hermano - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a editar - La herramienta muestra los datos asociados a dicho DNI - El usuario edita los campos que necesita editar
Escenario alternativo	No aplica

Tabla 5.38: CU-038: Editar Hermano

CU-039	<i>Editar Hermano</i>
Actores	Administrador.
Tipo de CU	Hermanos
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El hermano a editar debe formar parte del sistema como <u>Persona</u> .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Hermano
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Hermanos - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Hermanos - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a editar - La herramienta muestra los datos asociados a dicho DNI - El usuario edita los campos que necesita editar
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.39: CU-038: Editar Hermano

CU-039	<i>Editar Hermano</i>
Actores	Administrador.
Tipo de CU	Hermanos
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El hermano a editar debe formar parte del sistema como <u>Persona</u> .
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Hermano
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Hermanos - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Hermano - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a editar - La herramienta muestra los datos asociados a dicho DNI - El usuario edita los campos que necesita editar
Escenario alternativo	El sistema no encuentra en la BD el DNI facilitado por el usuario.

Tabla 5.40: CU-039: Editar Hermano

5.2.27. Casos de uso - Eliminar Hermano

CU-040	<i>Eliminar Hermano</i>
Actores	Administrador.
Tipo de CU	Hermanos
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El hermano a eliminar debe existir en el sistema como Hermano.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Hermano
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Hermanos - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Hermanos - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a Eliminar y hace click en Eliminar - La herramienta elimina los registros asociados en cascada de dicho DNI
Escenario alternativo	No aplica

Tabla 5.41: CU-040: Eliminar Hermano

CU-041	<i>Eliminar Hermano</i>
Actores	Administrador.
Tipo de CU	Hermanos
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El hermano a eliminar debe existir en el sistema como Hermano.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Hermano
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Hermanos - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Hermanos - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a Eliminar y hace click en Eliminar - La herramienta elimina los registros asociados en cascada de dicho DNI
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.42: CU-041: Eliminar Hermano

CU-042	<i>Eliminar Hermano</i>
Actores	Administrador.
Tipo de CU	Hermanos
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El hermano a eliminar debe existir en el sistema como Hermano.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Hermano
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Hermanos - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Hermanos - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a Eliminar y hace click en Eliminar - La herramienta elimina los registros asociados en cascada de dicho DNI
Escenario alternativo	El sistema no encuentra en la BD el DNI facilitado por el usuario.

Tabla 5.43: CU-042: Eliminar Hermano

5.2.28. CU-033: Listado de Control de Salida

CU-043	<i>Listado de Control de Salida</i>
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Listado de Control de Salida.
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Control de Salida - La herramienta gestiona el evento y realiza la navegabilidad
Escenario alternativo	No aplica

Tabla 5.44: CU-043: Listado de Control de Salida

5.2.29. Casos de uso - Exportar a PDF el Listado de Control de Salida

CU-044	<i>Exportar a PDF el Listado de Control de Salida</i>
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta exporta un documento PDF con la información del Listado
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Control de Salida - La herramienta gestiona el evento y realiza la navegabilidad - El usuario hace click en el icono de PDF situado en el listado - La herramienta recoge la información del listado y exporta un PDF para el usuario
Escenario alternativo	No aplico

Tabla 5.45: CU-044: Exportar a PDF el Listado de Control de Salida

5.2.30. Casos de uso - Exportar a Excel el Listado de Control de Salida

CU-045	<i>Exportar a Excel el Listado de Control de Salida</i>
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta exporta un documento Excel con la información del Listado
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Listado de Control de Salida - La herramienta gestiona el evento y realiza la navegabilidad - El usuario hace click en el icono de Excel situado en el listado - La herramienta recoge la información del listado y exporta un Excel para el usuario
Escenario alternativo	No aplico

Tabla 5.46: CU-045: Exportar a Excel el Listado de Control de Salida

5.2.31. Casos de uso - Añadir Control de Salida

CU-046	<i>Añadir Control de Salida</i>
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El hermano a darle de alta el control de salida debe existir en el sistema.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Añadir Control de Salida
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Añadir Control de Salida - La herramienta gestiona el evento y realiza la navegabilidad - El usuario cumplimenta el formulario
Escenario alternativo	No aplico

Tabla 5.47: CU-046: Añadir Control de Salida

CU-047	<i>Añadir Control de Salida</i>
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El hermano a dar de alta el control de salida debe existir en el sistema.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Añadir Control de Salida
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Añadir Control de Salida - La herramienta gestiona el evento y realiza la navegabilidad - El usuario cumplimenta el formulario
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.48: CU-047: Añadir Control de Salida

5.2.32. Casos de uso - Editar Control de Salida

CU-048	<i>Editar Control de Salida</i>
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El control de salida a editar debe pertenecer a un hermano dado de alta en el sistema.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Control de Salida
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Control de Salida - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a editar - La herramienta muestra los datos asociados a dicho DNI - El usuario edita los campos que necesita editar
Escenario alternativo	No aplica

Tabla 5.49: CU-048: Editar Control de Salida

CU-049	<i>Editar Control de Salida</i>
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El control de salida a editar debe pertenecer a un hermano dado de alta en el sistema.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Control de Salida
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Control de Salida - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a editar - La herramienta muestra los datos asociados a dicho DNI - El usuario edita los campos que necesita editar
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.50: CU-049: Editar Control de Salida

CU-050	<i>Editar Control de Salida</i>
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador. El control de salida a editar debe pertenecer a un hermano dado de alta en el sistema.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Editar Control de Salida
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Editar Control de Salida - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a editar - La herramienta muestra los datos asociados a dicho DNI - El usuario edita los campos que necesita editar
Escenario alternativo	El sistema no encuentra en la BD el DNI facilitado por el usuario.

Tabla 5.51: CU-050: Editar Control de Salida

5.2.33. Casos de uso - Eliminar Control de Salida

CU-051	<i>Eliminar Control de Salida</i>
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Control de Salida
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Control de Salida - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a Eliminar y hace click en Eliminar - La herramienta elimina los registros asociados en cascada de dicho DNI
Escenario alternativo	No aplica

Tabla 5.52: CU-051: Eliminar Control de Salida

CU-052	<i>Eliminar Control de Salida</i>
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Control de Salida
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Control de Salida - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a Eliminar y hace click en Eliminar - La herramienta elimina los registros asociados en cascada de dicho DNI
Escenario alternativo	El usuario no introduce correctamente los valores en el formulario. La herramienta detalla los motivos de los errores en el mismo.

Tabla 5.53: CU-052: Eliminar Control de Salida

CU-053	<i>Eliminar Control de Salida</i>
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Eliminar Control de Salida
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Eliminar Control de Salida - La herramienta gestiona el evento y realiza la navegabilidad - El usuario introduce el DNI a Eliminar y hace click en Eliminar - La herramienta elimina los registros asociados en cascada de dicho DNI
Escenario alternativo	El sistema no encuentra en la BD el DNI facilitado por el usuario.

Tabla 5.54: CU-053: Eliminar Control de Salida

CU-054	Estadística
Actores	Administrador.
Tipo de CU	Control de Salida
Precondiciones	El usuario debe haber iniciado sesión en la herramienta y tener rol Administrador.
Postcondiciones	La herramienta hace navegación hasta la pantalla de Estadística
Escenario	<ul style="list-style-type: none"> - El usuario hace click en el menú Control de Salida - La herramienta despliega las opciones pertenecientes al menú - El usuario hace click en Estadística - La herramienta gestiona el evento y realiza la navegabilidad - El sistema muestra una pantalla con las diferentes gráficas obtenidas
Escenario alternativo	- No aplica

Tabla 5.55: CU-054: Estadística

5.3. Diagrama de Secuencias

Los diagramas de secuencia (DS) representan las interacciones entre el usuario y el sistema para cada acción que se puede realizar en dicho sistema. Este tipo de diagrama es útil para complementar los casos de uso realizados anteriormente, donde de forma visual, se muestra como el usuario hace uso de la herramienta y la respuesta de esta misma ante esa interacción. A continuación se desglosan los diferentes diagramas de secuencia para la herramienta para cada caso de uso citado anteriormente.

5.3.1. Diagrama de Secuencia - Entrar

Diagrama de secuencia asociado al Caso de Uso CU001

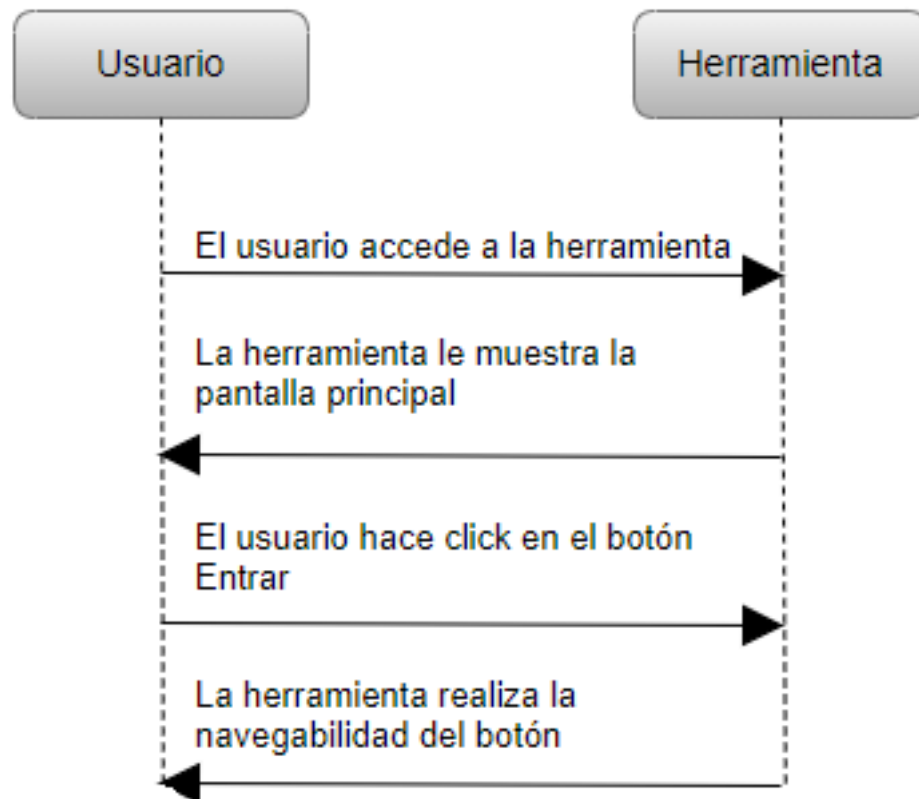


Figura 5.4: DS01 - Entrar

5.3.2. Diagrama de Secuencia - Iniciar Sesión

Diagrama de secuencia asociado al Caso de Uso CU002

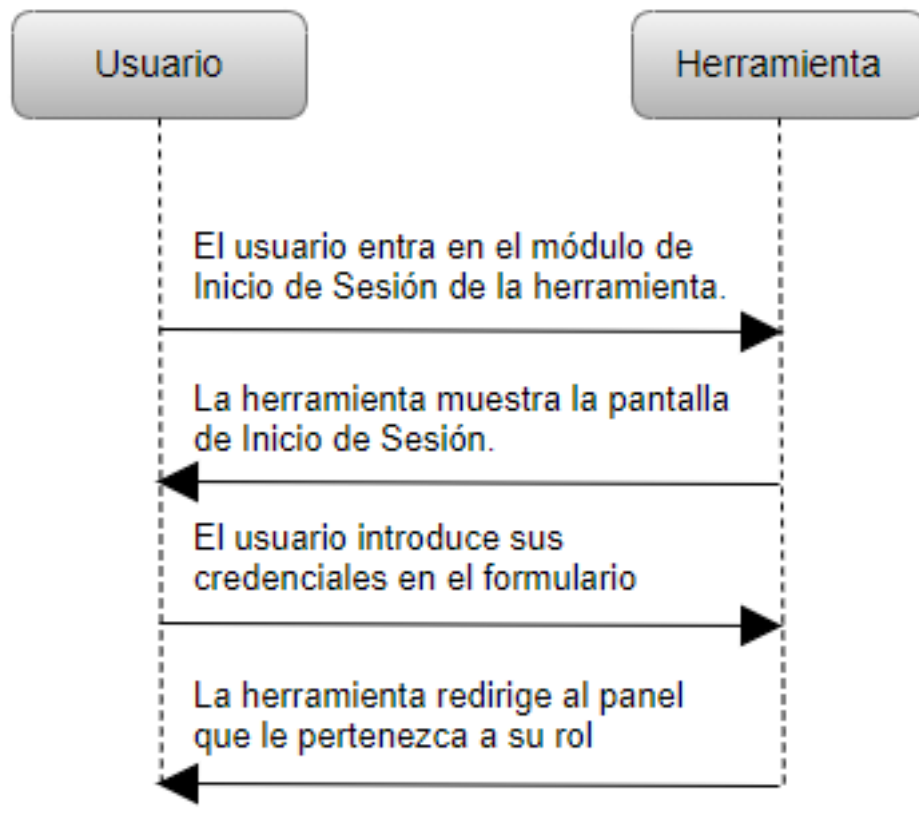


Figura 5.5: DS02 - Iniciar Sesión

5.3.3. Diagrama de Secuencia - Personas

Diagrama de secuencia asociado al Caso de Uso CU003

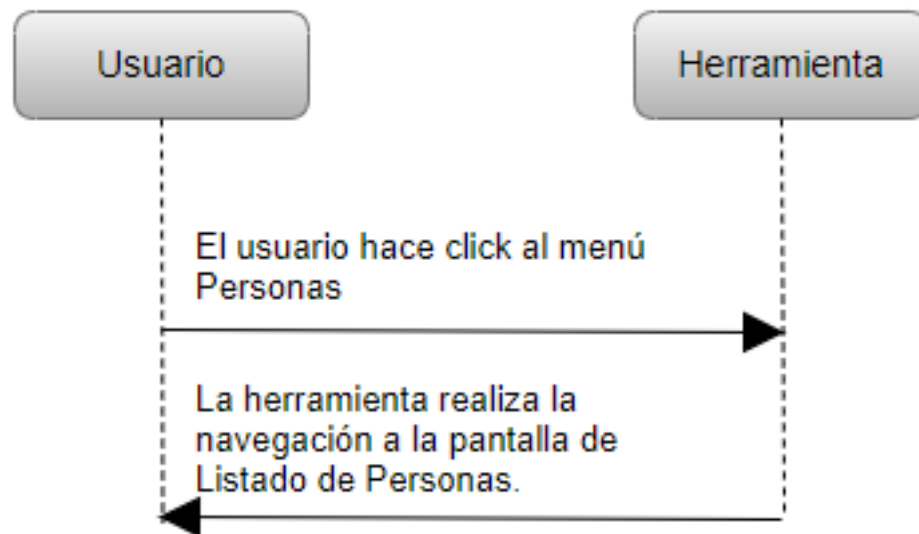


Figura 5.6: DS03 - Personas

5.3.4. Diagrama de Secuencia - Control de Salida

Diagrama de secuencia asociado al Caso de Uso CU004

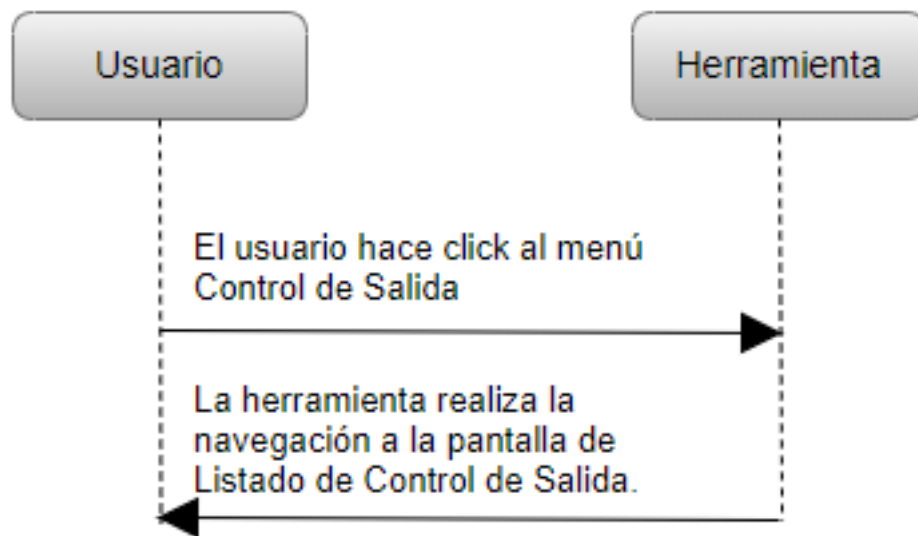


Figura 5.7: DS04 - Control de Salida

5.3.5. Diagrama de Secuencia - Hermanos

Diagrama de secuencia asociado al Caso de Uso CU005

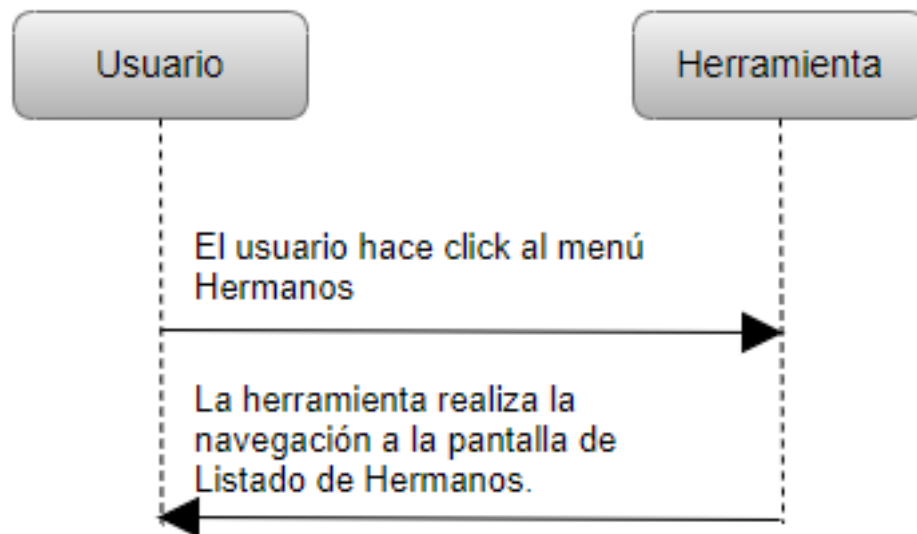


Figura 5.8: DS05 - Hermanos

5.3.6. Diagrama de Secuencia - Túnicas

Diagrama de secuencia asociado al Caso de Uso CU006

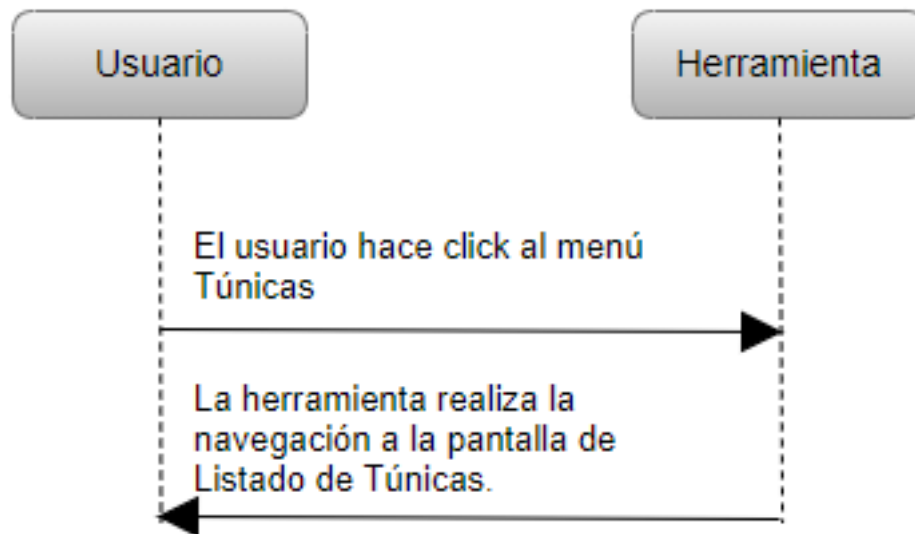


Figura 5.9: DS06 - Túnicas

5.3.7. Diagrama de Secuencia - Reserva tu Control de Salida

Diagrama de secuencia asociado al Caso de Uso CU007

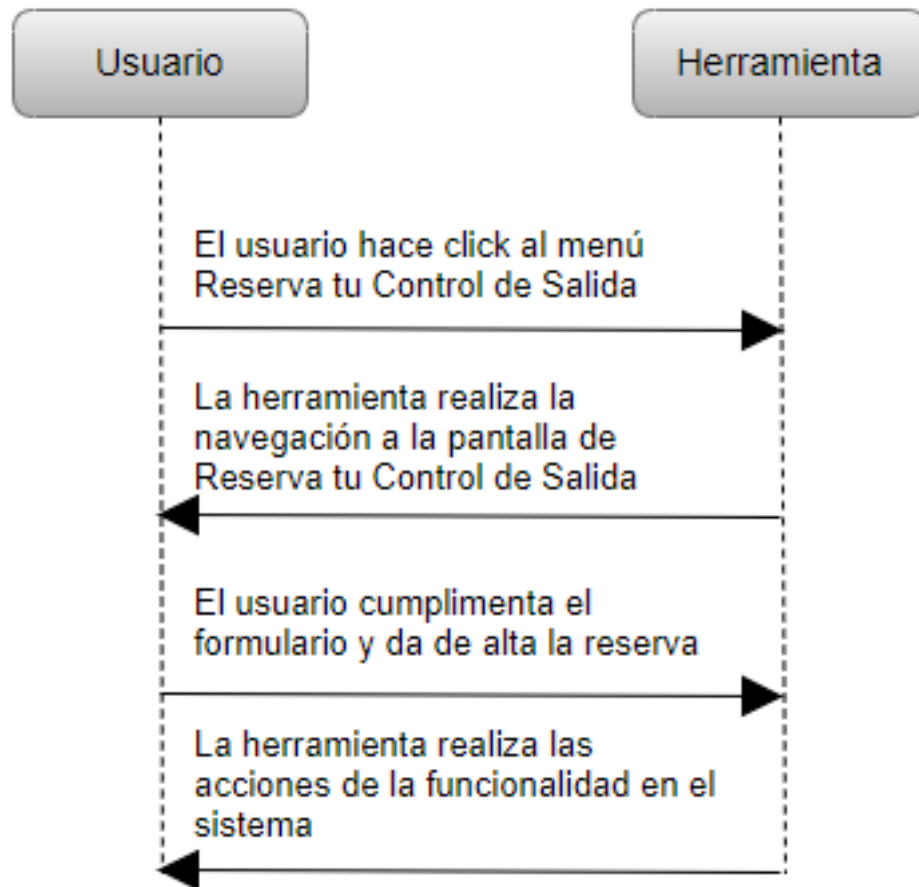


Figura 5.10: DS07 - Reserva tu Control de Salida

5.3.8. Diagrama de Secuencia - Tus cuotas

Diagrama de secuencia asociado al Caso de Uso CU009

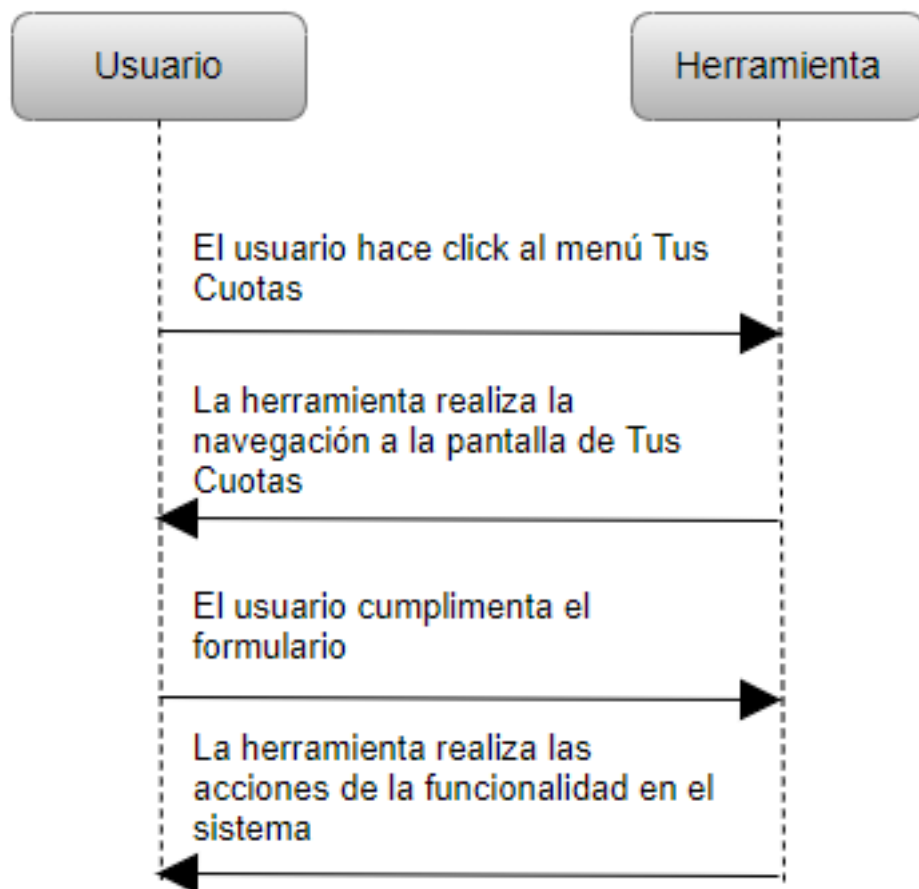


Figura 5.11: DS09 - Tus Cuotas

5.3.9. Diagrama de Secuencia - Listado de Personas

Diagrama de secuencia asociado al Caso de Uso CU011

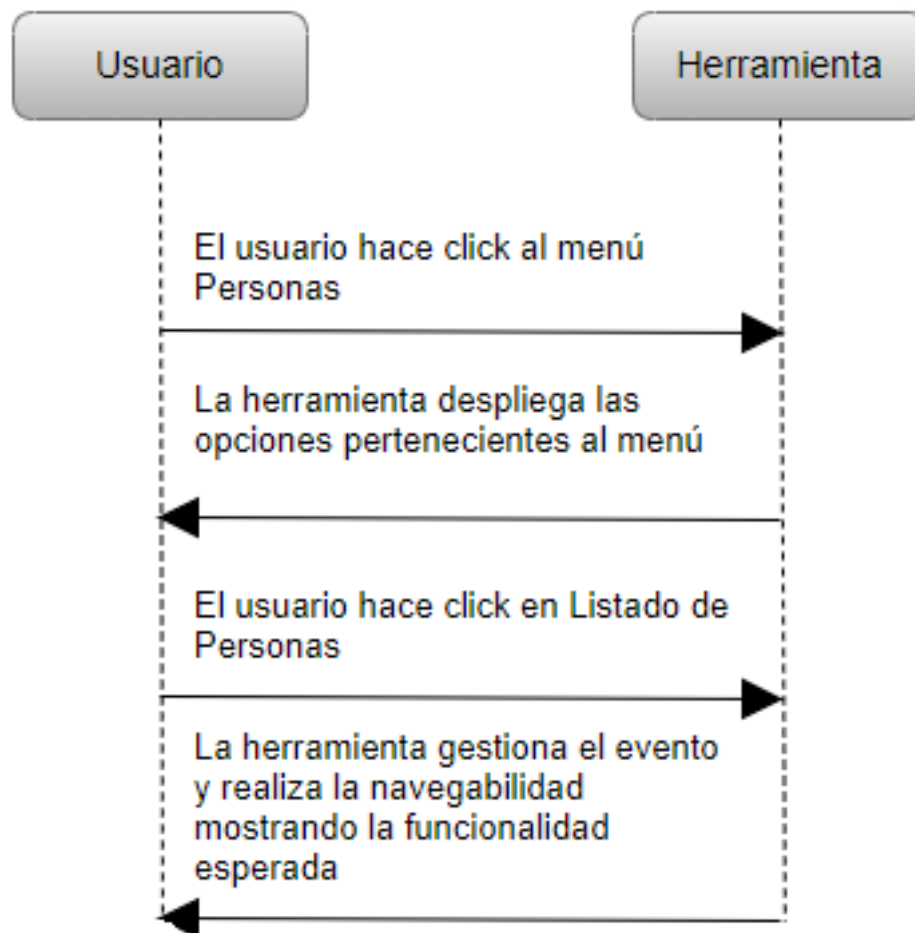


Figura 5.12: DS10 - Listado de Personas

5.3.10. Diagrama de Secuencia - Exportar a PDF el Listado de Personas

Diagrama de secuencia asociado al Caso de Uso CU012

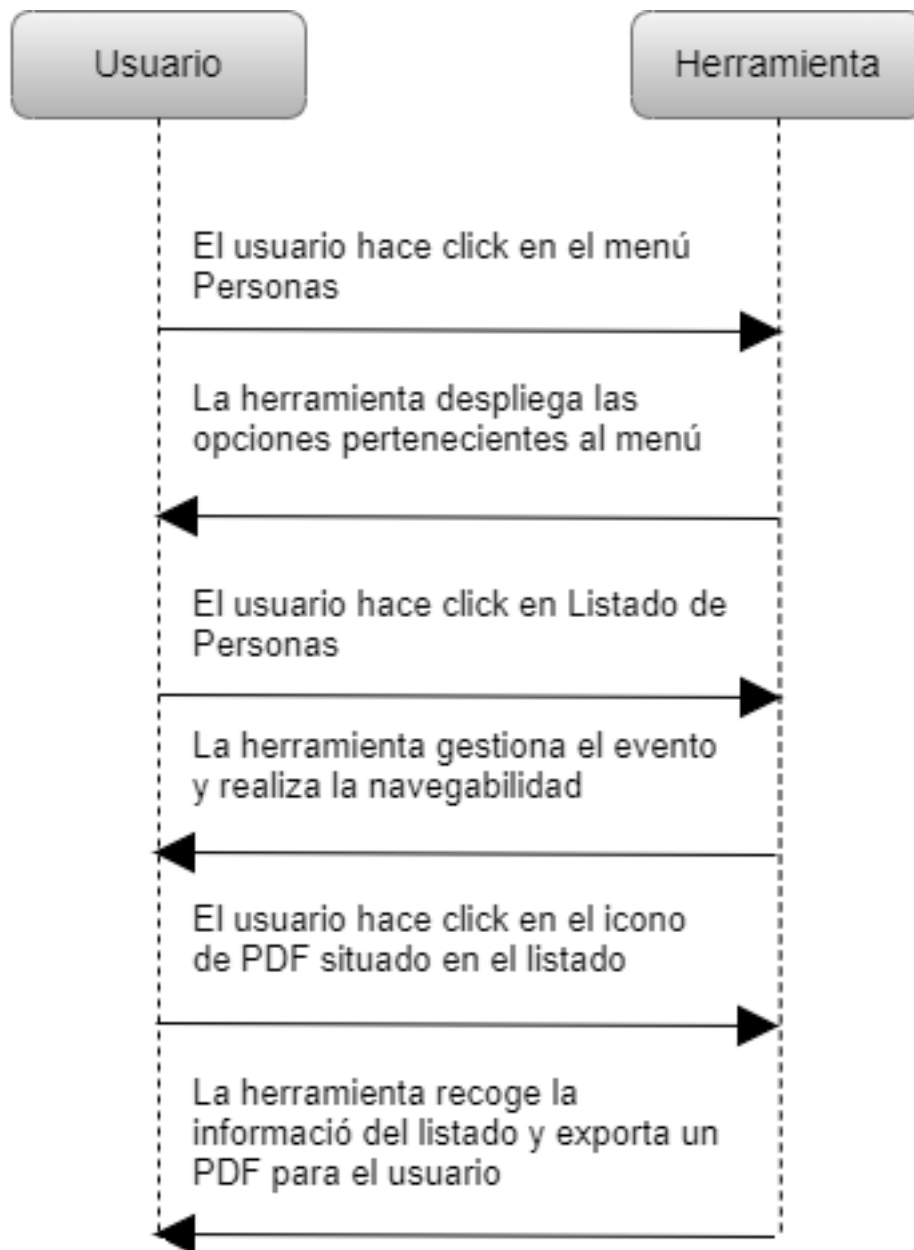


Figura 5.13: DS11 - Listado de Personas a PDF

5.3.11. Diagrama de Secuencia - Exportar a Excel el Listado de Personas

Diagrama de secuencia asociado al Caso de Uso CU013

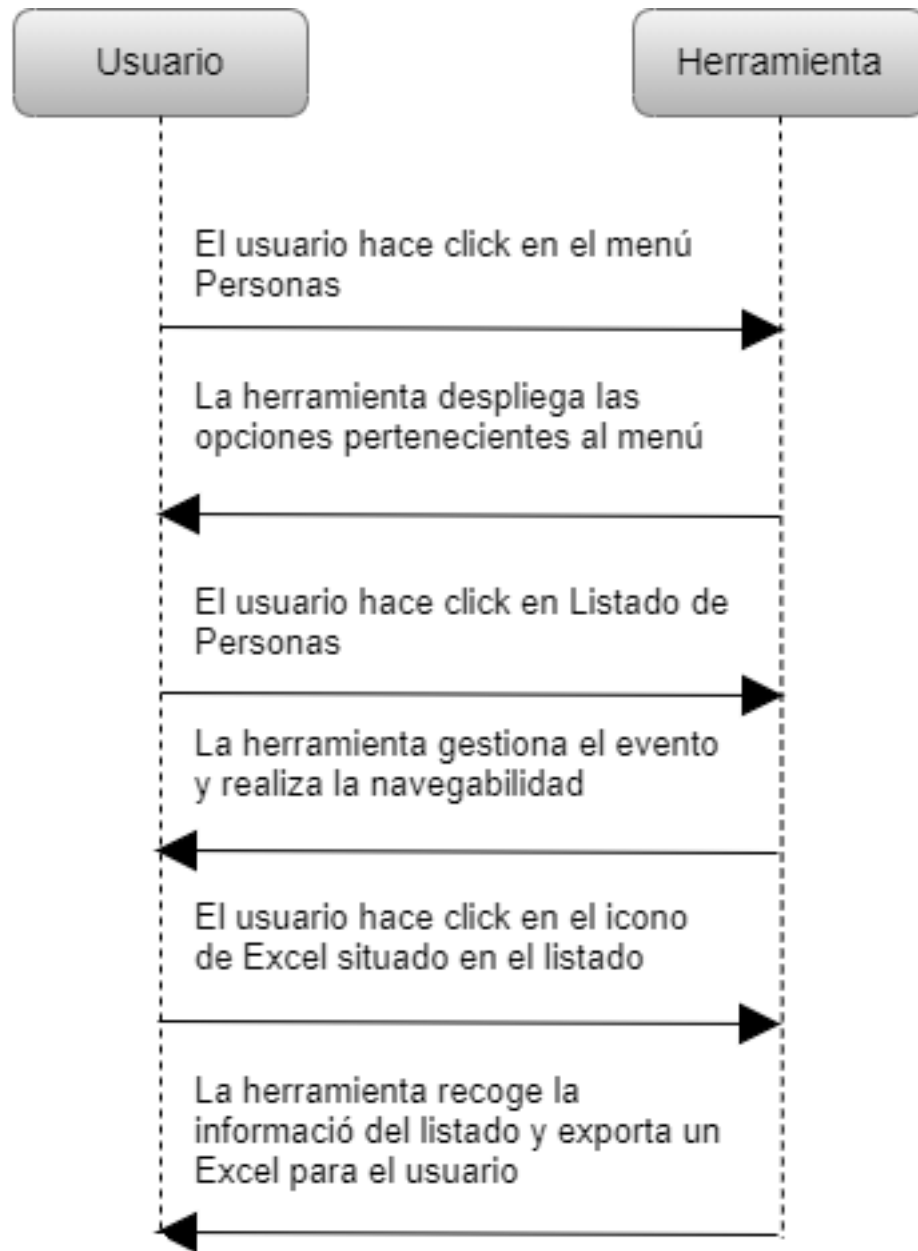


Figura 5.14: DS12 - Listado de Personas a Excel

5.3.12. Diagrama de Secuencia - Añadir Persona

Diagrama de secuencia asociado al Caso de Uso CU014

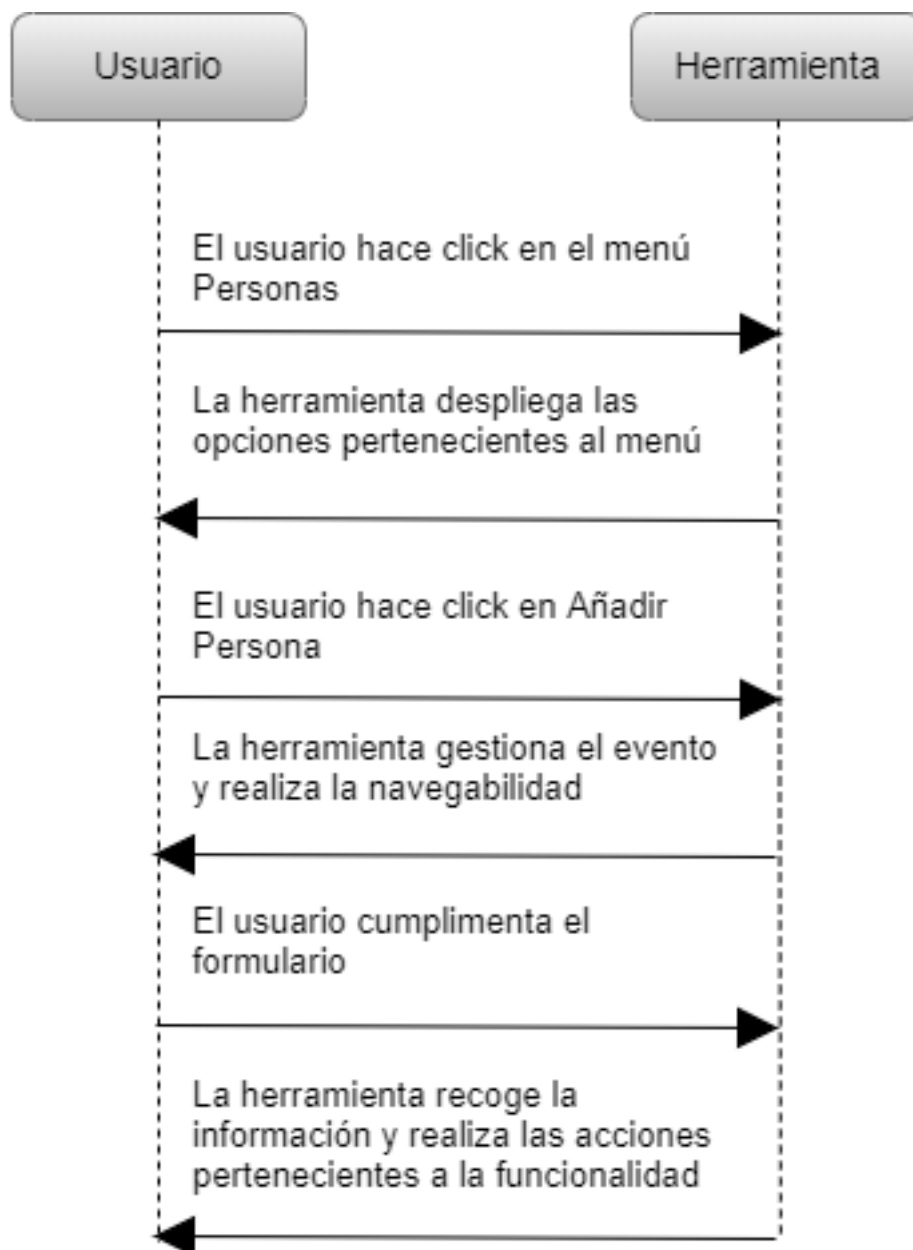


Figura 5.15: DS13 - Añadir Persona

5.3.13. Diagrama de Secuencia - Editar Persona

Diagrama de secuencia asociado al Caso de Uso CU016

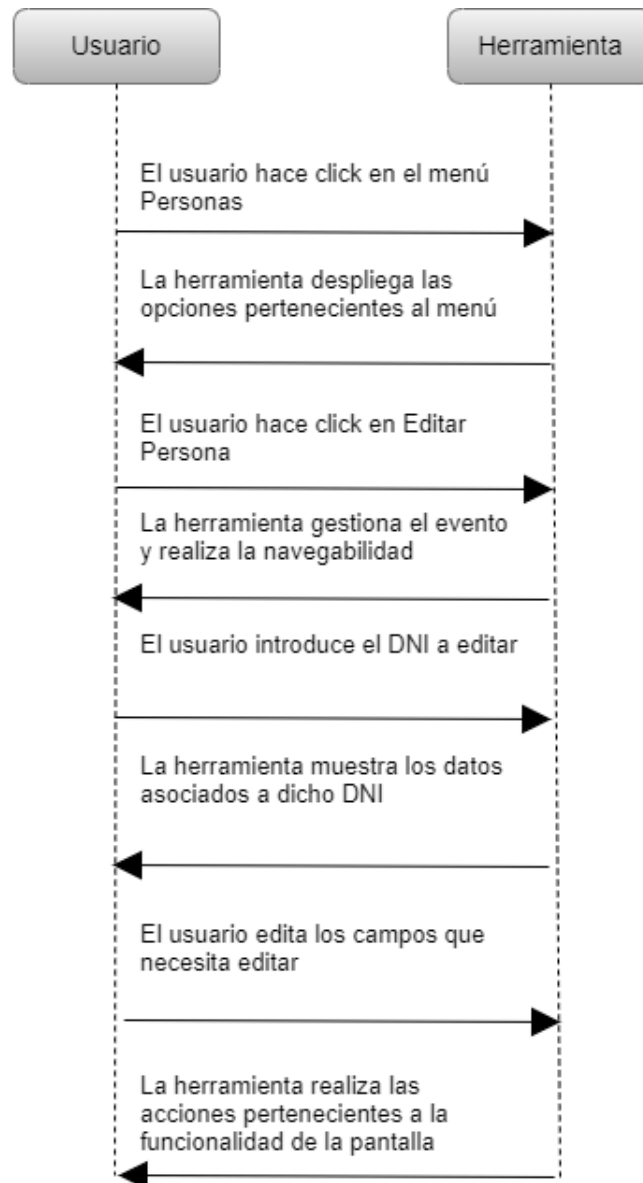


Figura 5.16: DS14 - Editar Persona

5.3.14. Diagrama de Secuencia - Eliminar Persona

Diagrama de secuencia asociado al Caso de Uso CU019

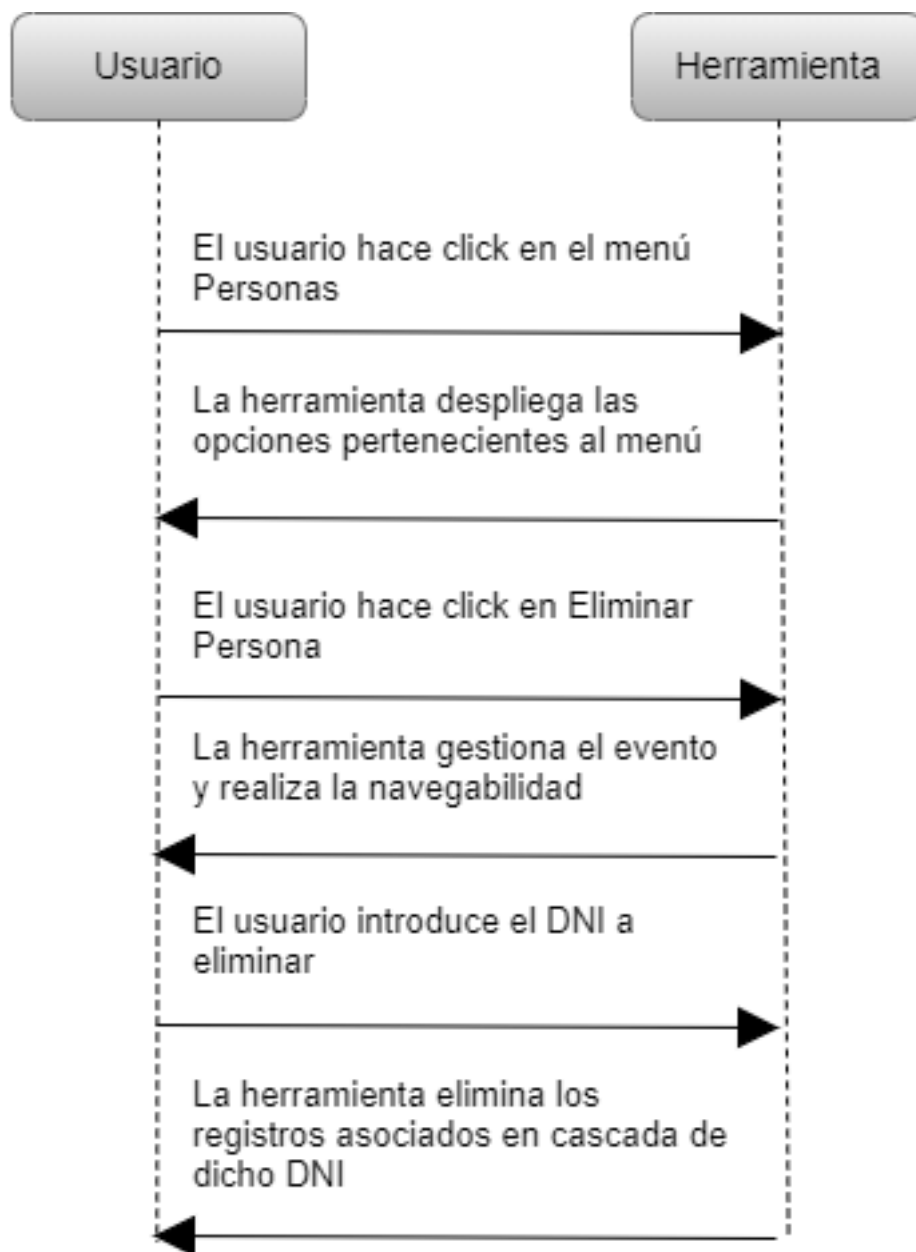


Figura 5.17: DS15 - Eliminar Persona

5.3.15. Diagrama de Secuencia - Listado de Túnicas

Diagrama de secuencia asociado al Caso de Uso CU022

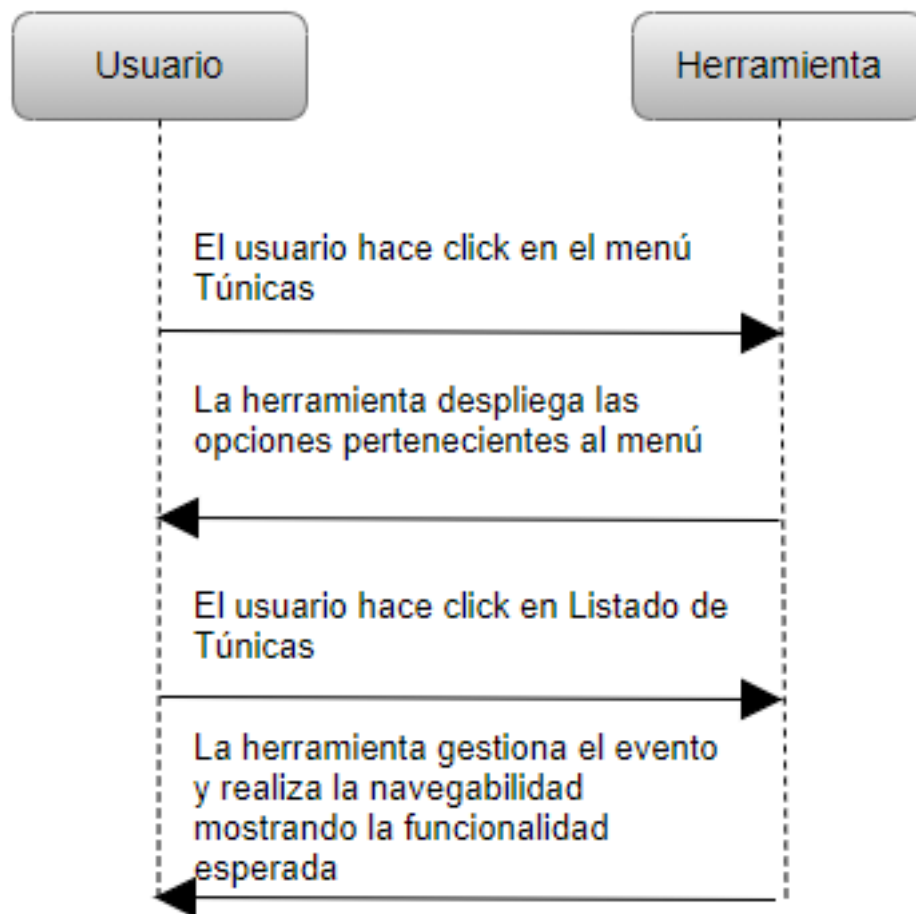


Figura 5.18: DS16 - Listado de Túnicas

5.3.16. Diagrama de Secuencia - Exportar a PDF el Listado de Túnicas

Diagrama de secuencia asociado al Caso de Uso CU023

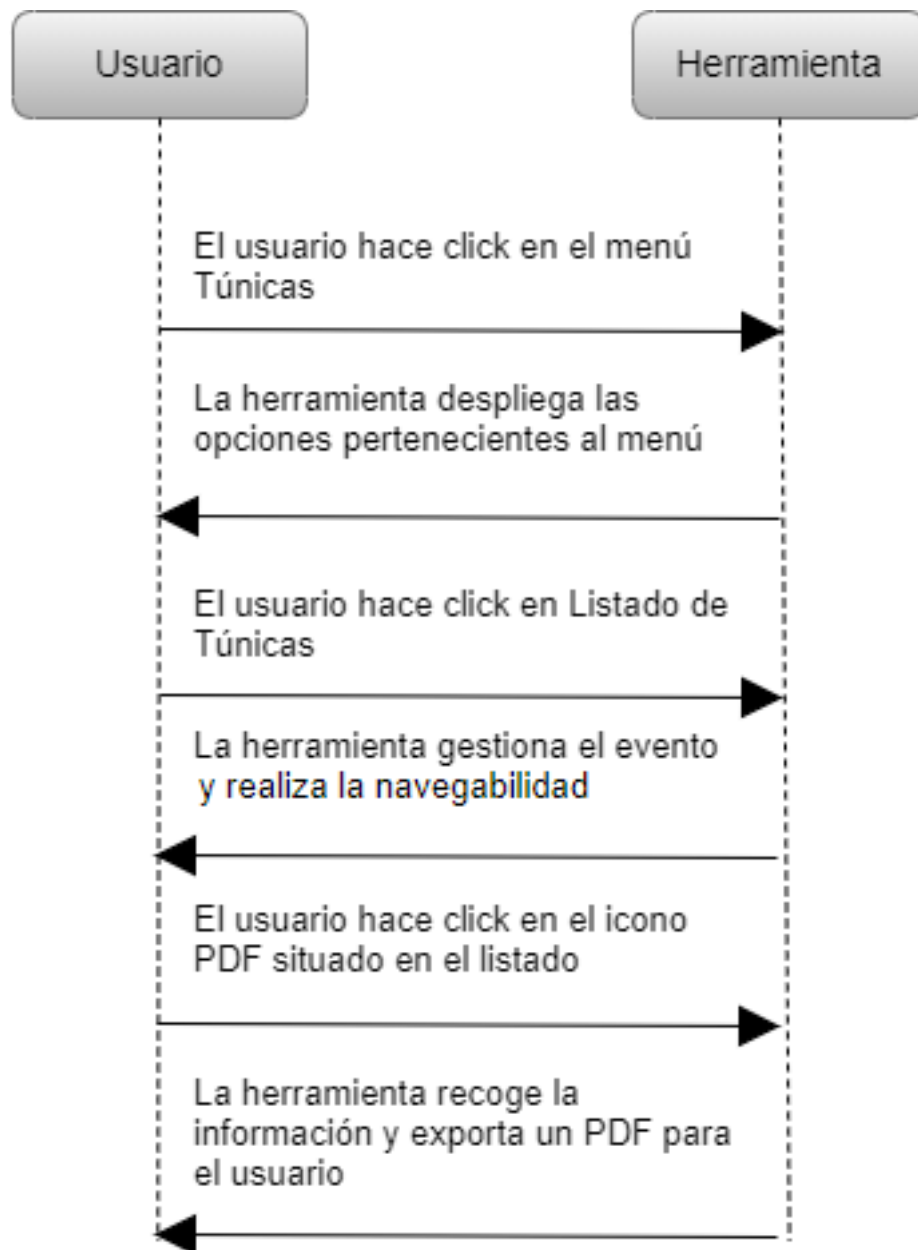


Figura 5.19: DS17 - Listado de Túnicas a PDF

5.3.17. Diagrama de Secuencia - Exportar a Excel el Listado de Túnicas

Diagrama de secuencia asociado al Caso de Uso CU024

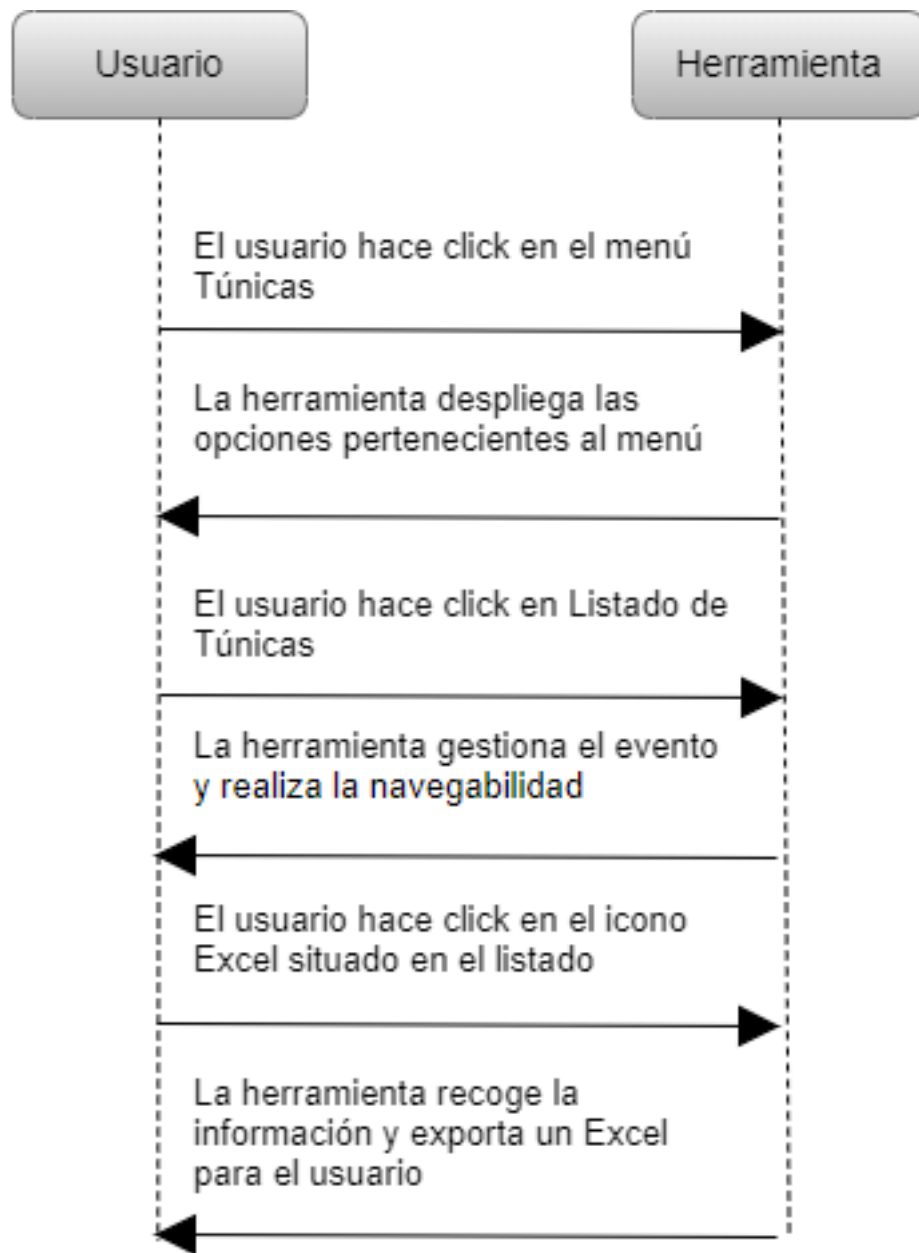


Figura 5.20: DS18 - Listado de Túnicas a Excel

5.3.18. Diagrama de Secuencia - Añadir Túnica

Diagrama de secuencia asociado al Caso de Uso CU025

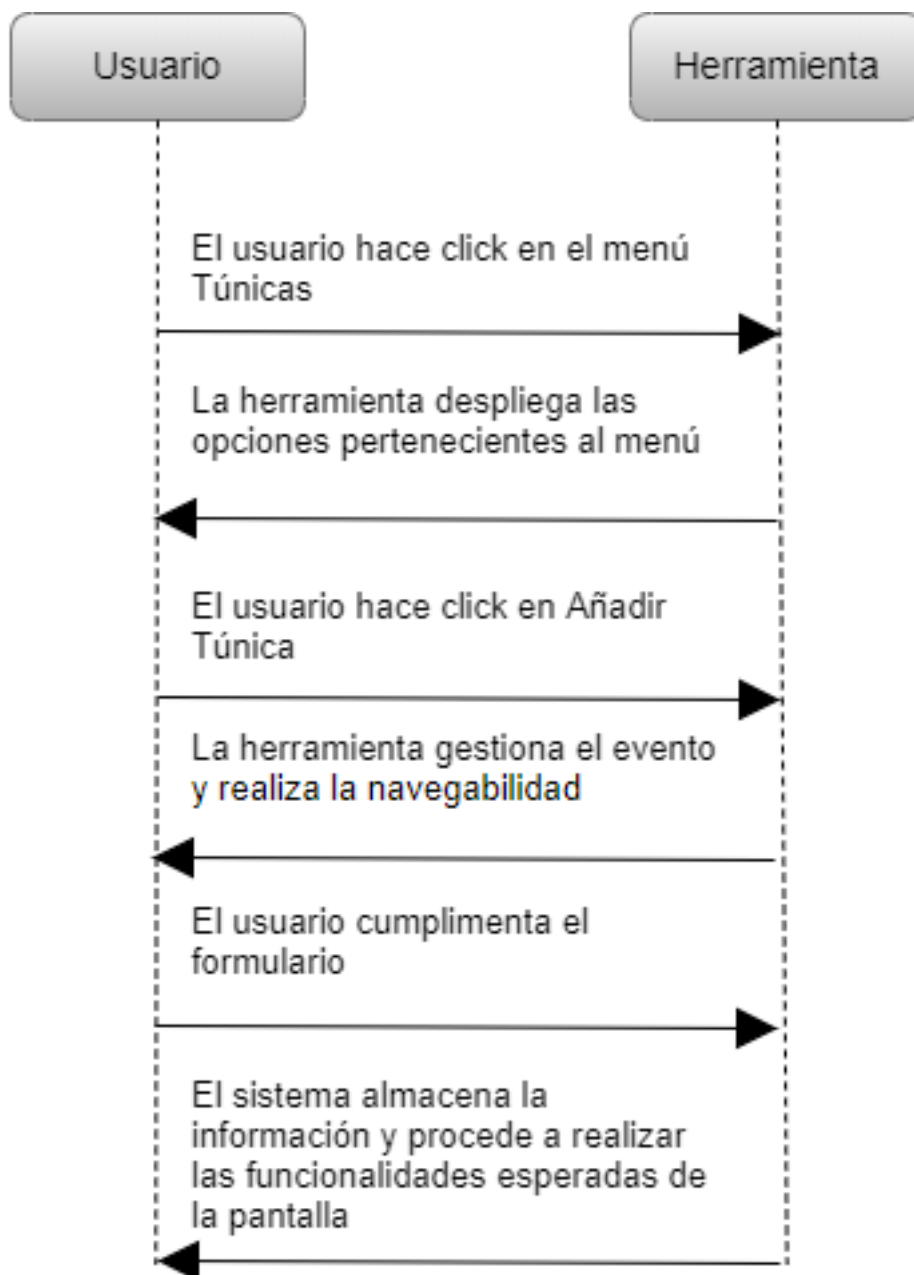


Figura 5.21: DS19 - Añadir Túnica

5.3.19. Diagrama de Secuencia - Editar Túnica

Diagrama de secuencia asociado al Caso de Uso CU027

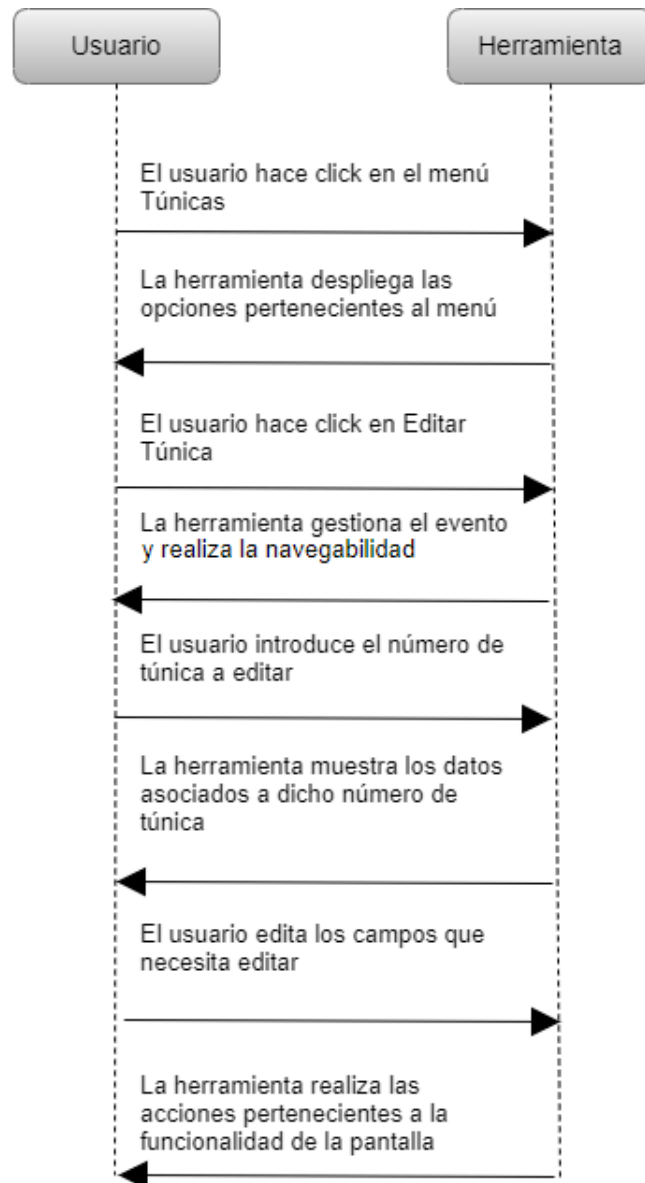


Figura 5.22: DS20 - Editar Túnica

5.3.20. Diagrama de Secuencia - Eliminar Túnica

Diagrama de secuencia asociado al Caso de Uso CU030

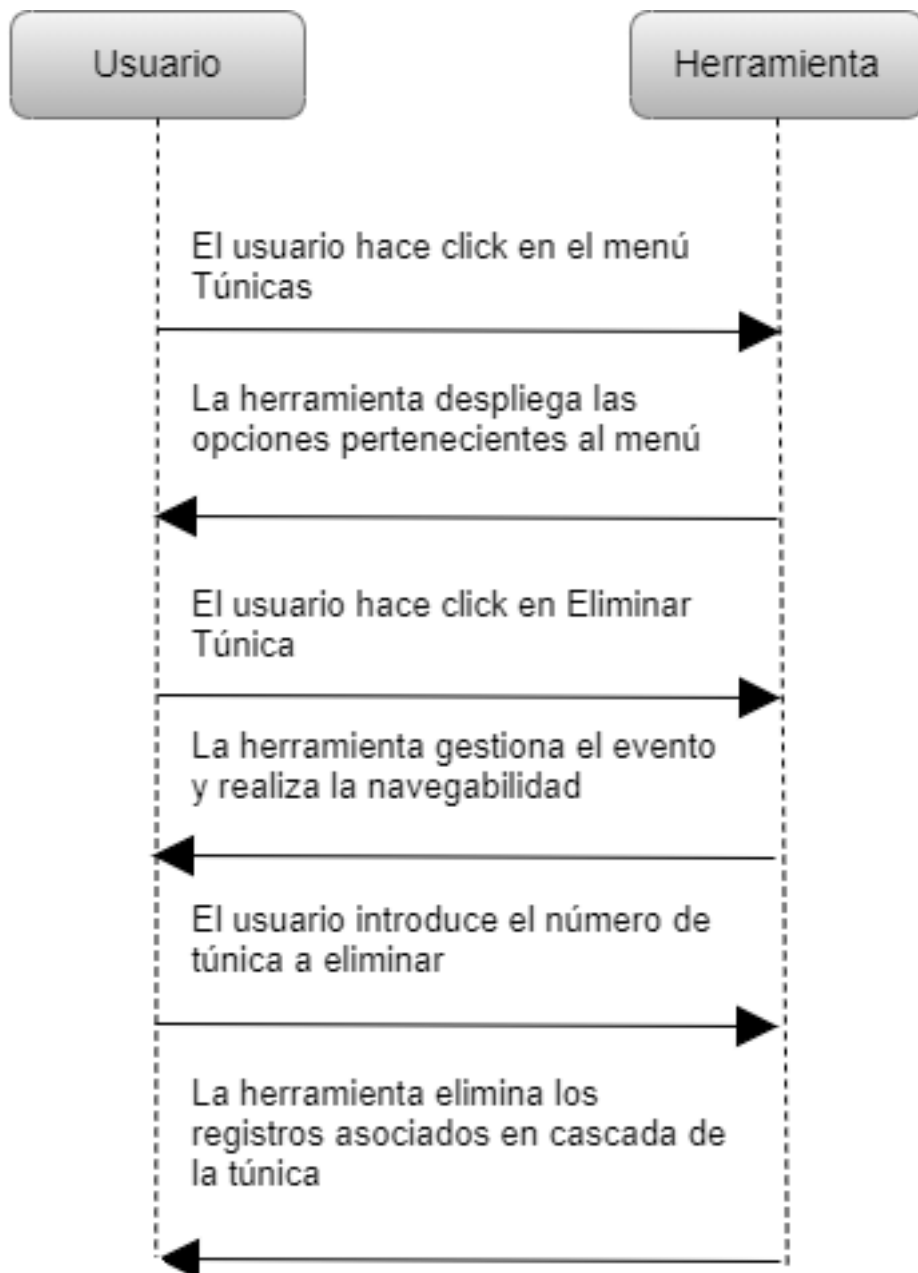


Figura 5.23: DS21 - Eliminar Túnica

5.3.21. Diagrama de Secuencia - Listado de Hermanos

Diagrama de secuencia asociado al Caso de Uso CU033

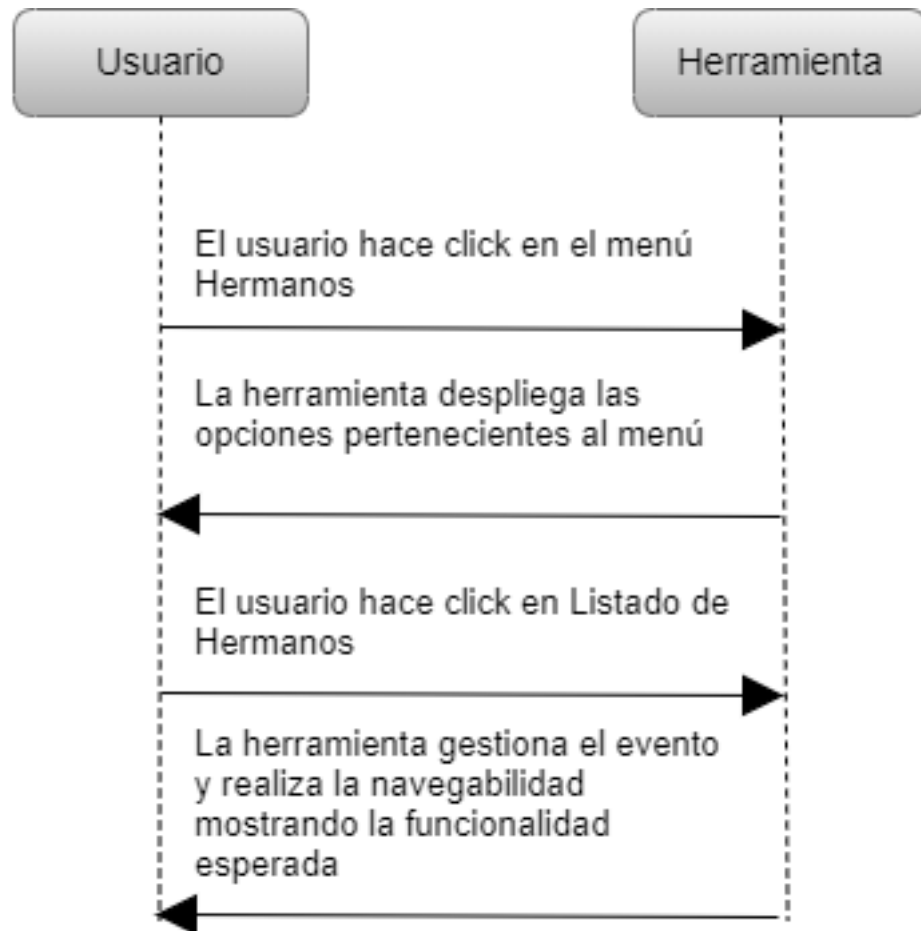


Figura 5.24: DS22 - Listado de Hermanos

5.3.22. Diagrama de Secuencia - Exportar a PDF el Listado de Hermanos

Diagrama de secuencia asociado al Caso de Uso CU034

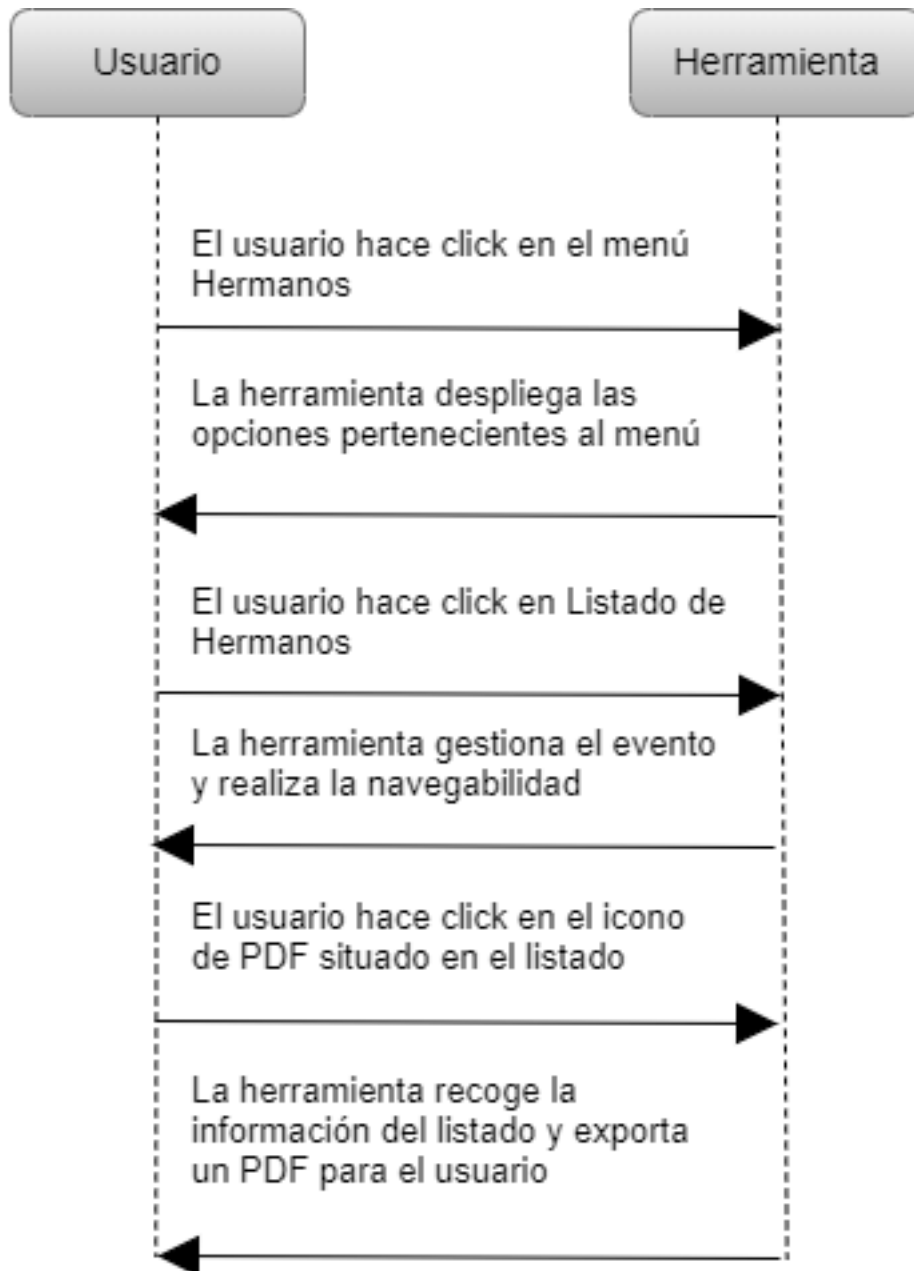


Figura 5.25: DS23 - Listado de Hermanos a PDF

5.3.23. Diagrama de Secuencia - Exportar a Excel el Listado de Hermanos

Diagrama de secuencia asociado al Caso de Uso CU035

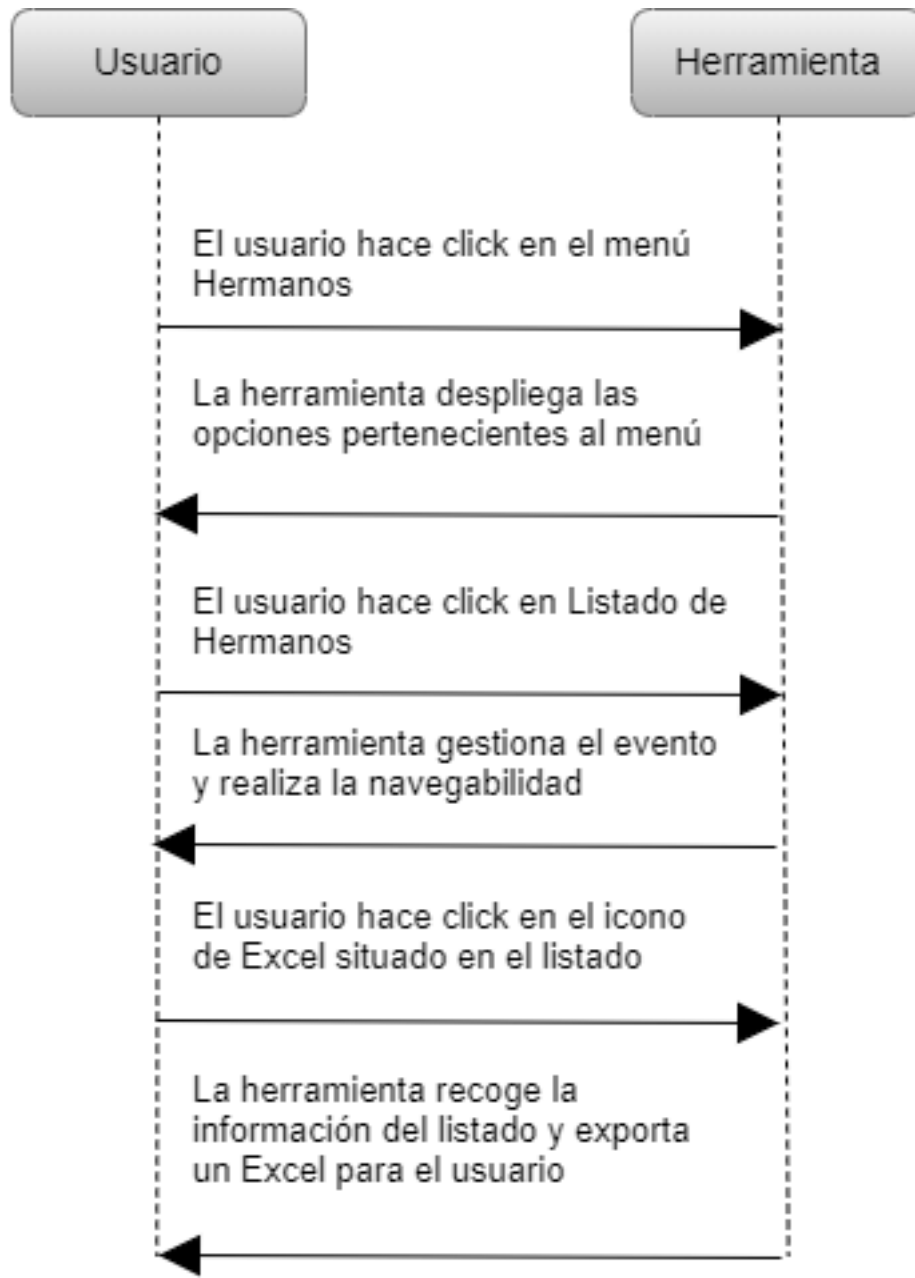


Figura 5.26: DS24 - Listado de Hermanos a Excel

5.3.24. Diagrama de Secuencia - Añadir Hermano

Diagrama de secuencia asociado al Caso de Uso CU036

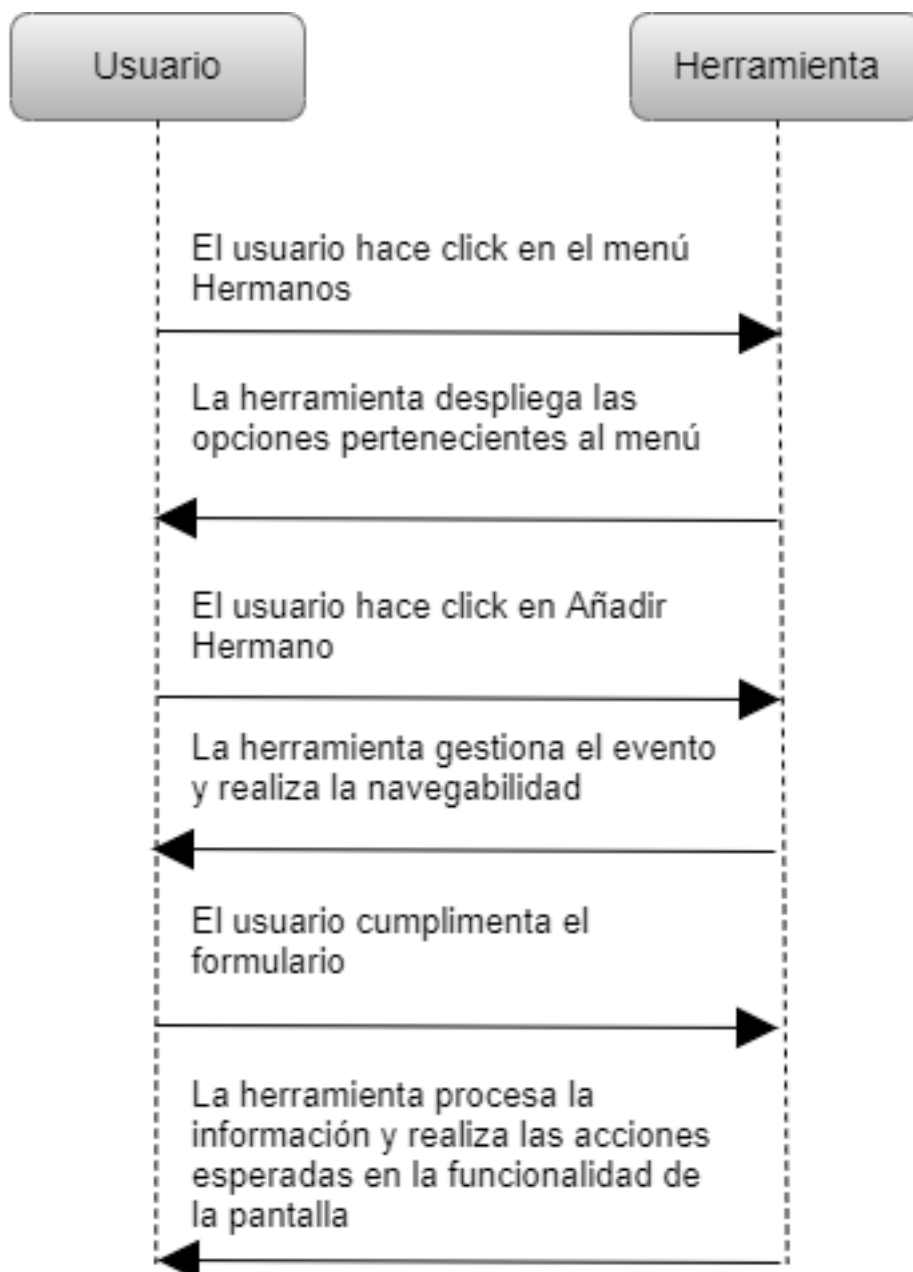


Figura 5.27: DS25 - Añadir Hermano

5.3.25. Diagrama de Secuencia - Editar Hermano

Diagrama de secuencia asociado al Caso de Uso CU038

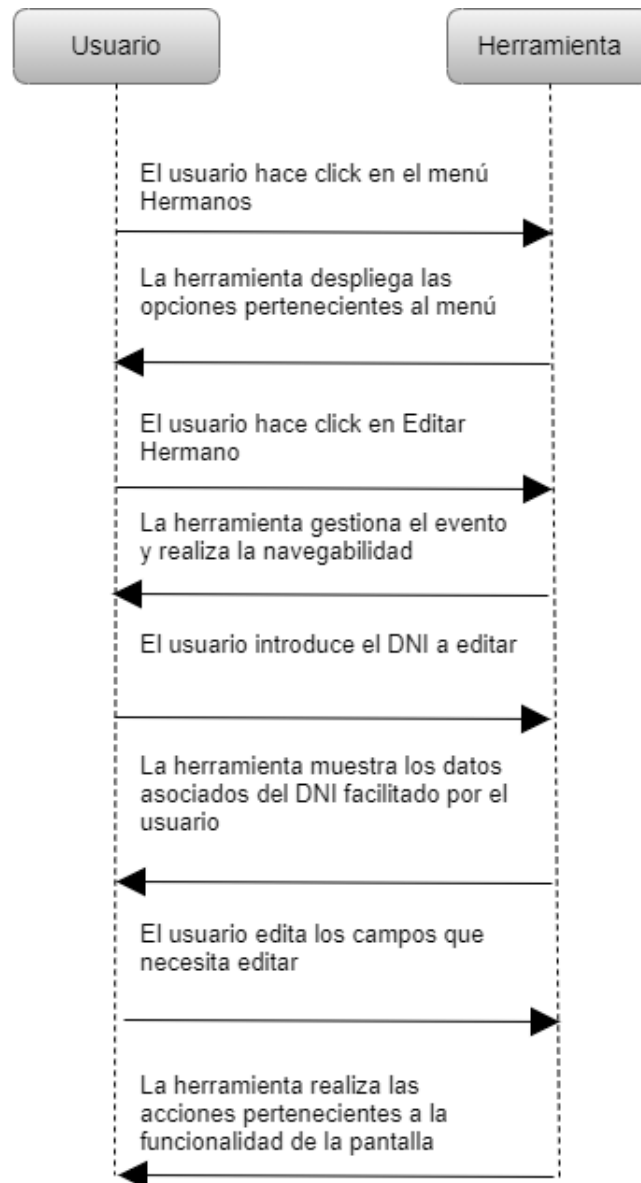


Figura 5.28: DS26 - Editar Hermano

5.3.26. Diagrama de Secuencia - Eliminar Hermano

Diagrama de secuencia asociado al Caso de Uso CU040

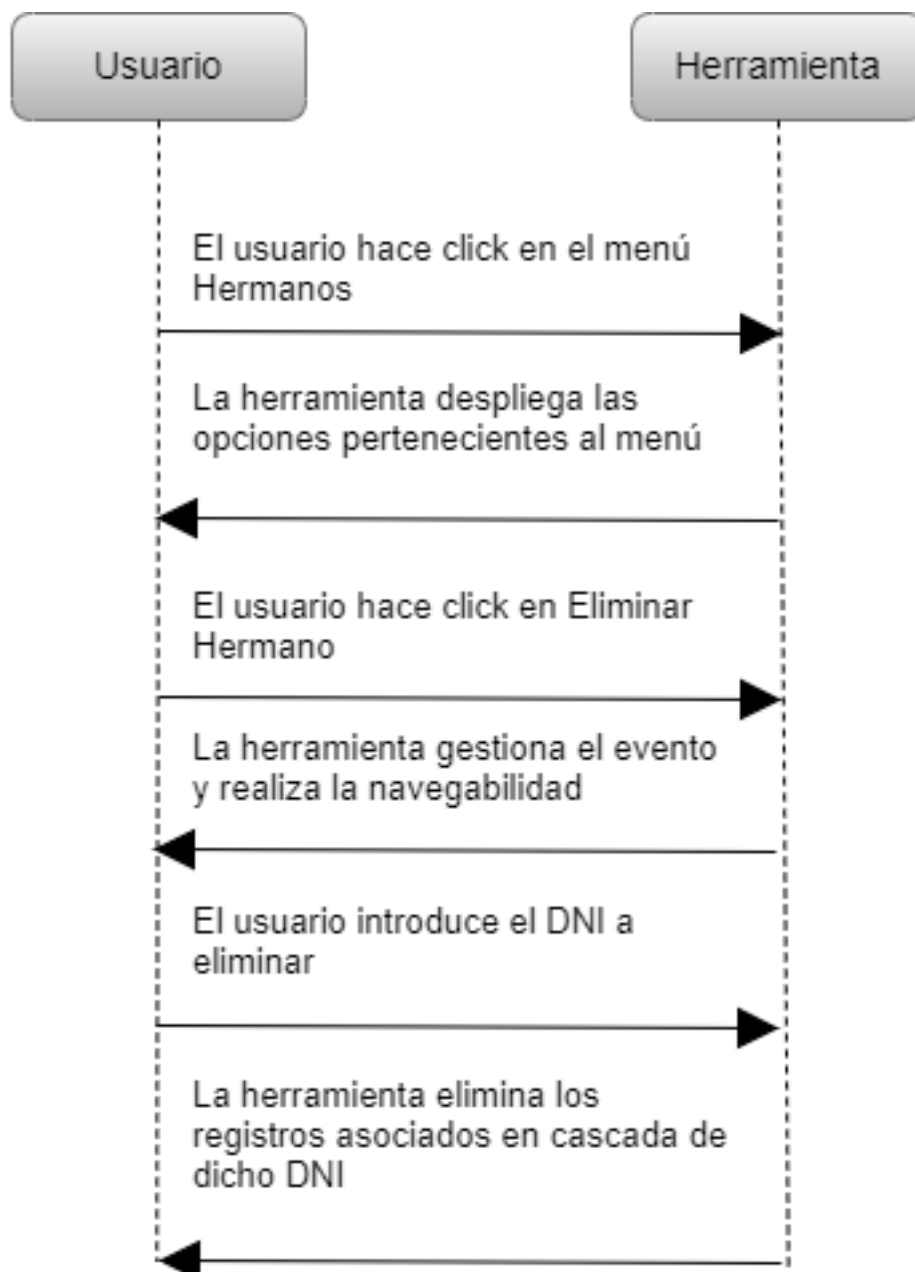


Figura 5.29: DS27 - Eliminar Hermano

5.3.27. Diagrama de Secuencia - Listado de Control de Salida

Diagrama de secuencia asociado al Caso de Uso CU043

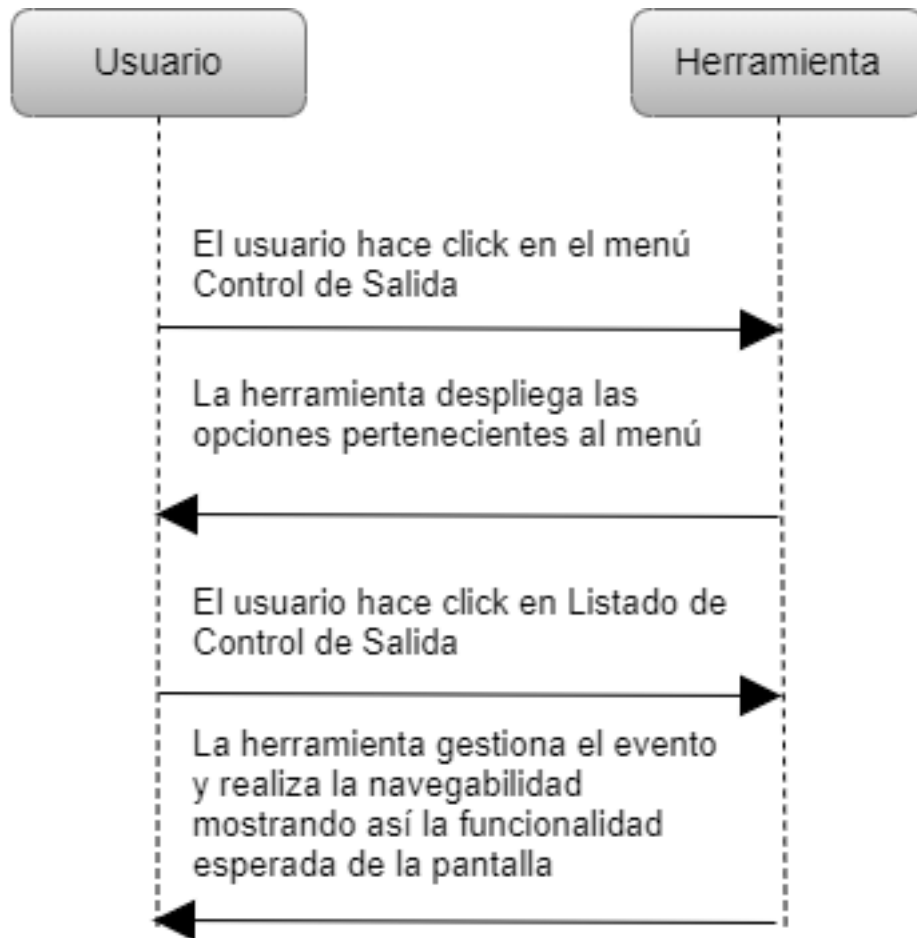


Figura 5.30: DS28 - Listado de Control de Salida

5.3.28. Diagrama de Secuencia - Exportar a PDF el Listado de Control de Salida

Diagrama de secuencia asociado al Caso de Uso CU044

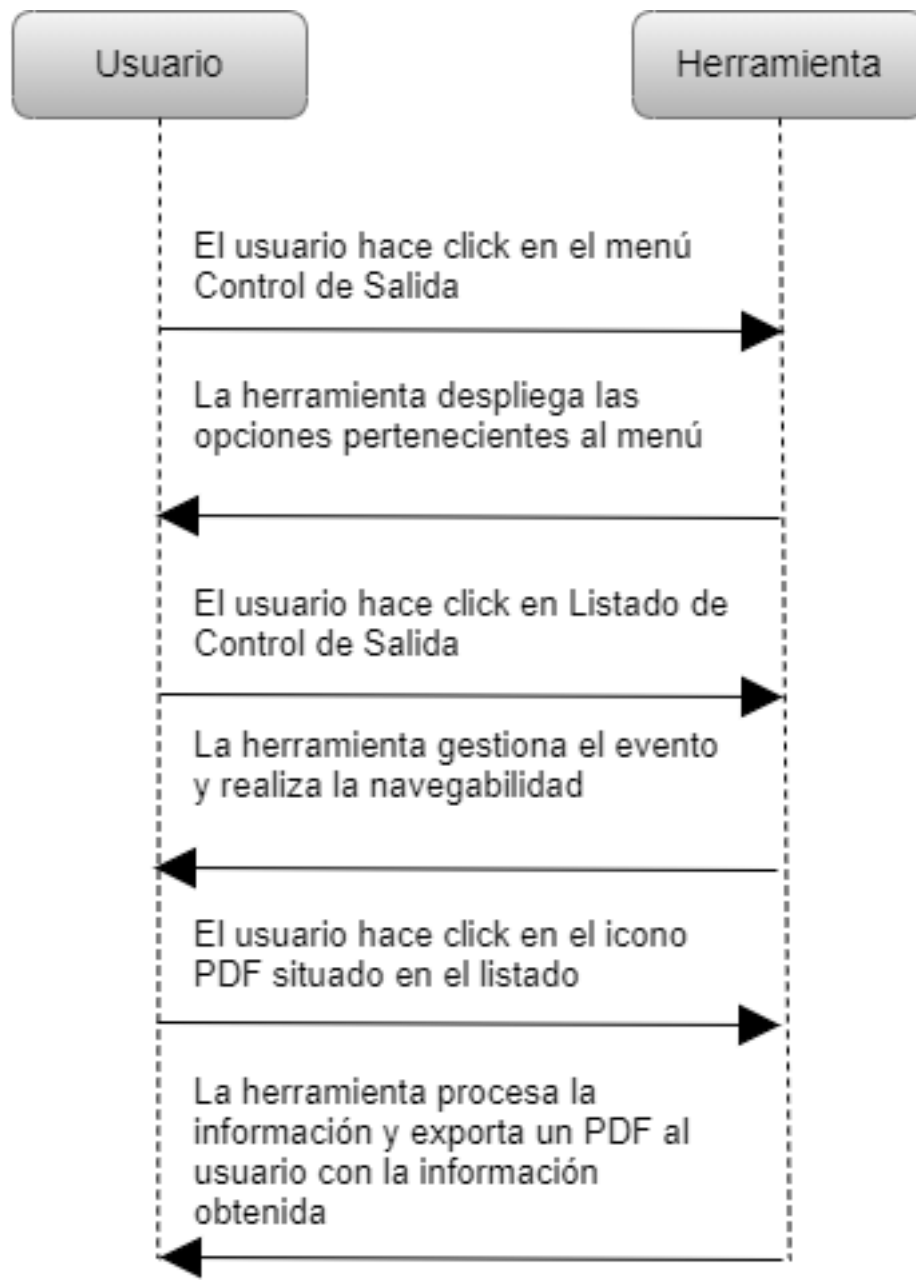


Figura 5.31: DS29 - Listado de Control de Salida a PDF

5.3.29. Diagrama de Secuencia - Exportar a Excel el Listado de Control de Salida

Diagrama de secuencia asociado al Caso de Uso CU045

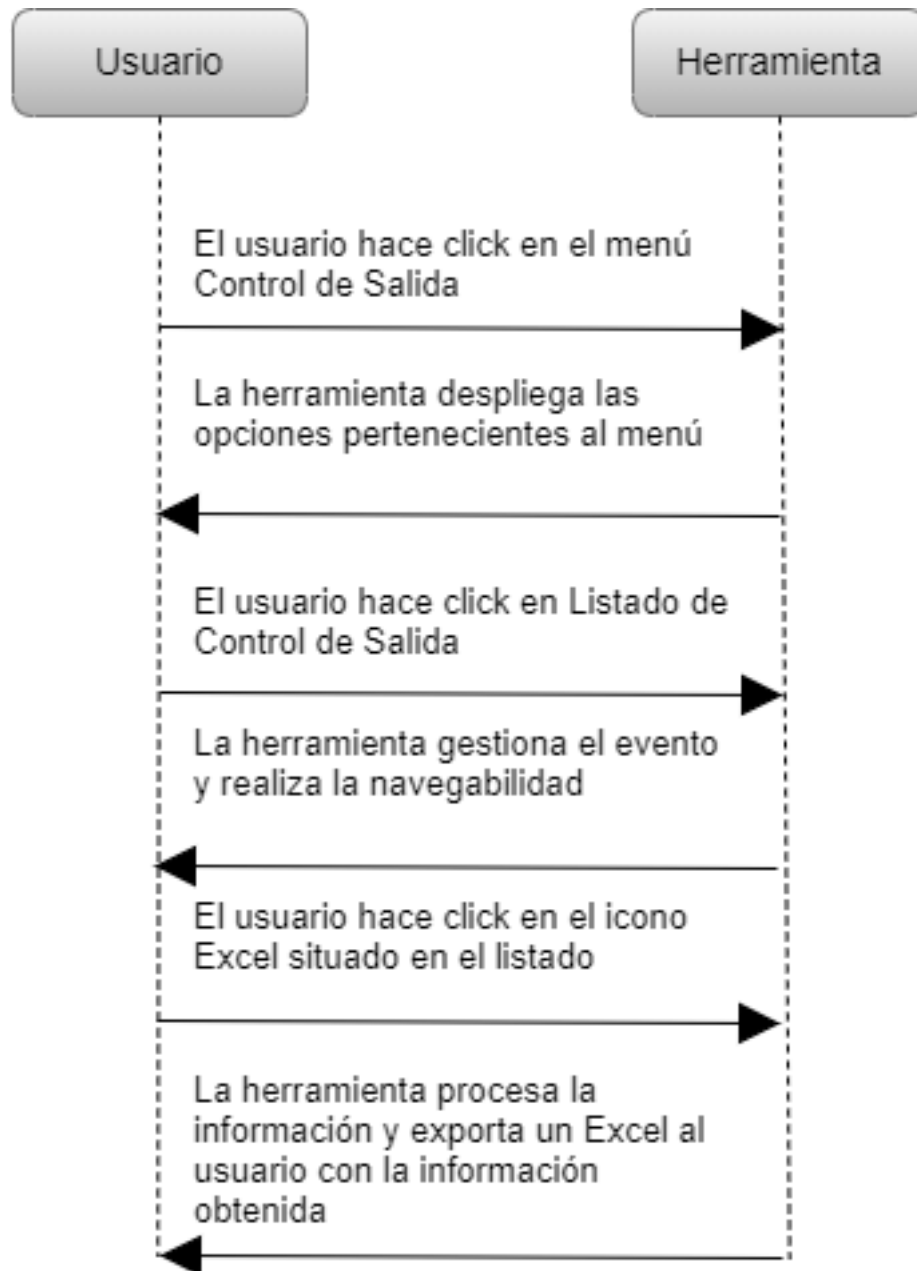


Figura 5.32: DS30 - Listado de Control de Salida a Excel

5.3.30. Diagrama de Secuencia - Añadir Control de Salida

Diagrama de secuencia asociado al Caso de Uso CU046

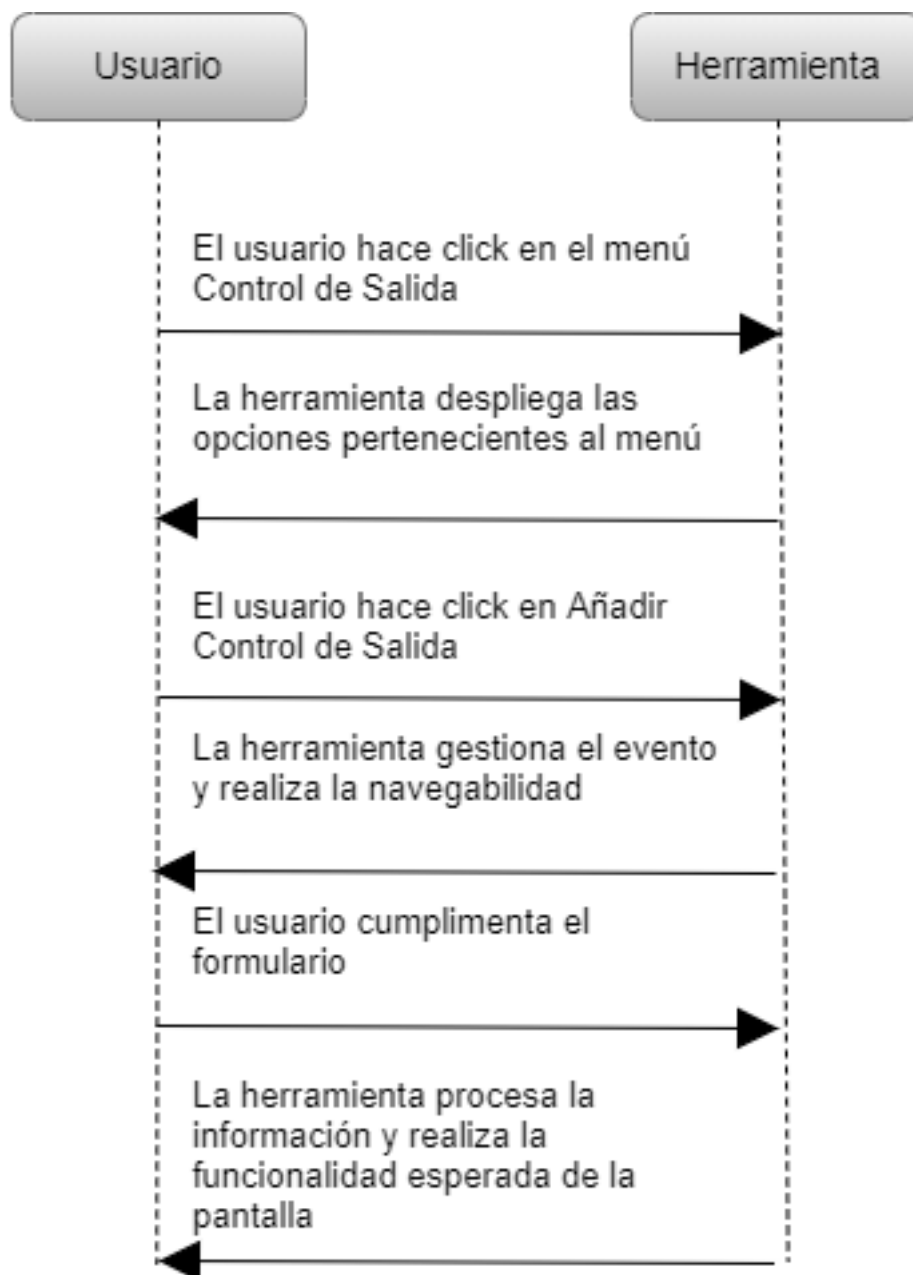


Figura 5.33: DS31 - Añadir Control de Salida

5.3.31. Diagrama de Secuencia - Editar Control de Salida

Diagrama de secuencia asociado al Caso de Uso CU048

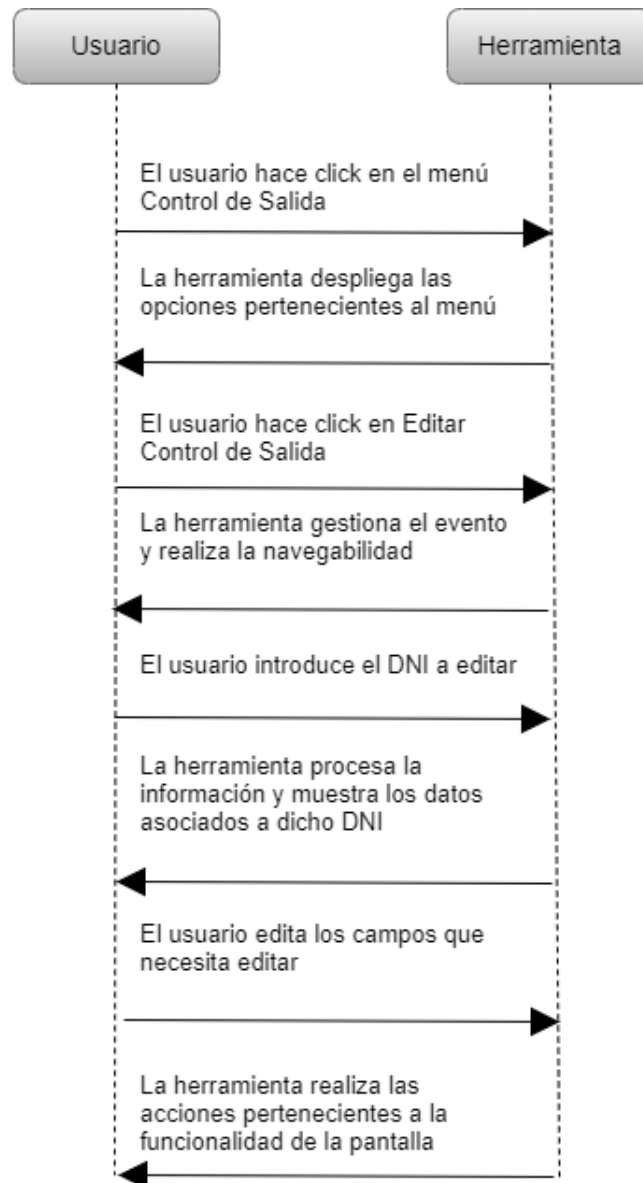


Figura 5.34: DS32 - Editar Control de Salida

5.3.32. Diagrama de Secuencia - Eliminar Control de Salida

Diagrama de secuencia asociado al Caso de Uso CU051

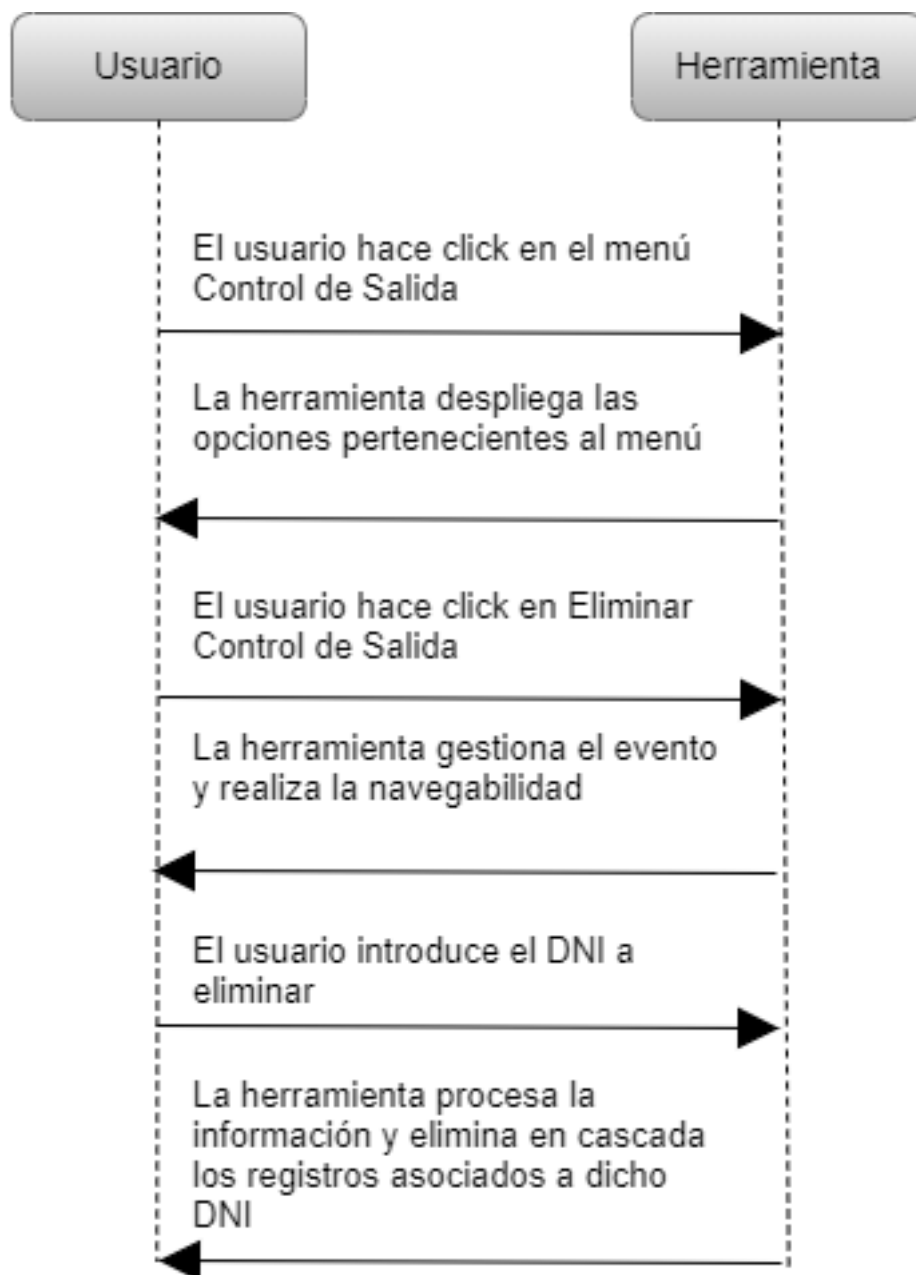


Figura 5.35: DS33 - Eliminar Control de Salida

5.3.33. Diagrama de Secuencia - Estadísticas

Diagrama de secuencia asociado al Caso de Uso CU054

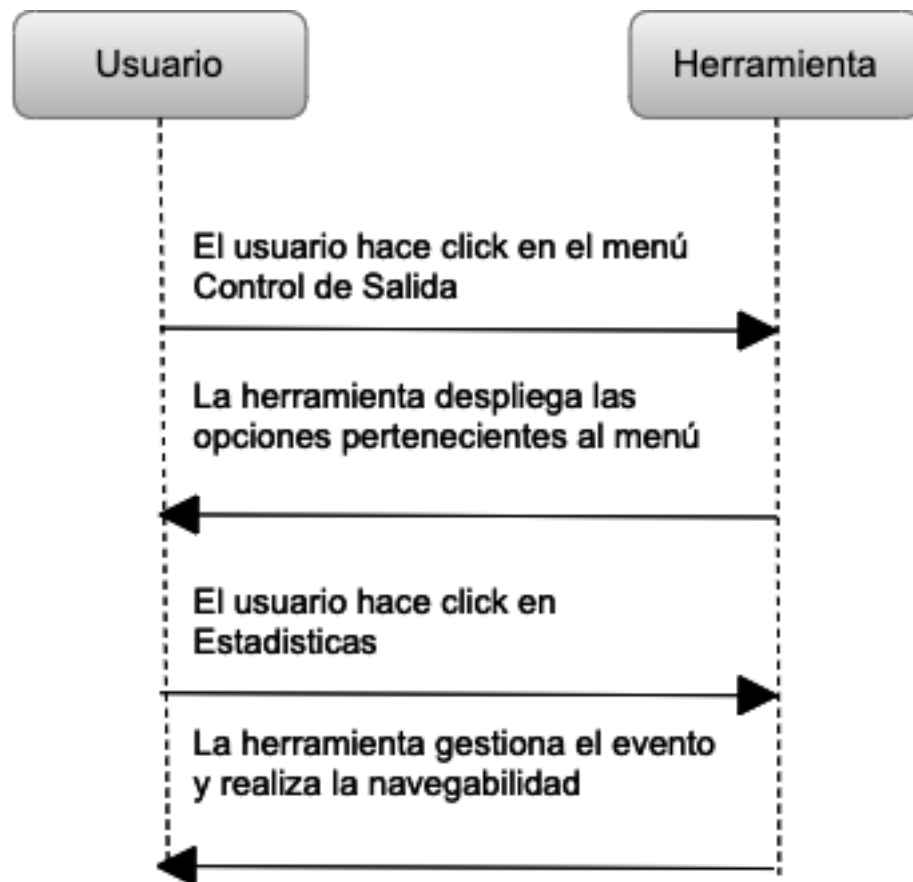


Figura 5.36: DS34 - Estadística

5.4. Modelo de clases inicial

Una vez realizado los diferentes análisis para la creación de la herramienta, se esboza un primer modelo de las clases que compondrán la aplicación para llevar a cabo las acciones descritas anteriormente. Este modelo no contará con los atributos internos para cada modelo que indicaremos a continuación pues servirá como toma de contacto hacia la implementación de la herramienta, conformando una buena estructura principal para el desarrollo e implementación en los futuros capítulos del documento. Así pues, la herramienta se compondrá de los siguientes modelos:

- **Control de Salida** Es una representación sobre el control que se gestiona el día que la Hermandad hace su salida procesional por lo que es necesario moderarla y tenerla en cuenta para el desarrollo
- **Hermano** Es una representación de la persona que pertenece a la Hermandad por lo que es necesario moderarla y tenerla en cuenta para el desarrollo
- **Ocupación** Es una representación del patrimonio que portan los hermanos en la salida procesional. Se gestiona como una especie de inventario por lo que es necesario moderarla y tenerla en cuenta para el desarrollo
- **Paso** Es una representación del acompañamiento al titular que hace el hermano durante su salida procesional por lo que es necesario moderarla y tenerla en cuenta para el desarrollo
- **Persona** Es una representación básica del hermano por lo que es necesario moderarla y tenerla en cuenta para el desarrollo
- **Procesión** Es una representación del día que la Hermandad hace su salida procesional por lo que es necesario moderarla y tenerla en cuenta para el desarrollo
- **Recibo** Es una representación del tipo de gestión realizada por la Hermandad al hermano por lo que es necesario moderarla y tenerla en cuenta para el desarrollo
- **Tipo de Pago** Es un diccionario sobre el módulo de Pago de por lo que es necesario moderarla y tenerla en cuenta para el desarrollo
- **Tipo de Recibo** Es un diccionario sobre el módulo de Recibo de por lo que es necesario moderarla y tenerla en cuenta para el desarrollo
- **Túnica** Es una representación de las túnicas que portan los hermanos ya que pertenecen a la Hermandad y se gestiona como una especie de inventario por lo que es necesario moderarla y tenerla en cuenta para el desarrollo
- **Usuario** Es una representación básica sobre los usuarios que tienen acceso a tunicAPP por lo que es necesario moderarla y tenerla en cuenta para el desarrollo

Finalmente, se expone un diagrama de clases el cual, para una mejor visualización, se expondrá en formato horizontal y en la siguiente página del documento.

Capítulo 6

Diseño

En este capítulo se describe el proceso de diseño de la herramienta. En concreto, se explican las decisiones de diseño que se han tomado en base a la funcionalidad requerida presentada en los capítulos anteriores y al uso que se espera hacer de la aplicación.

6.1. Diseño de la herramienta

Una vez realizado el estudio previo para la creación de la herramienta en el capítulo anterior, y tras haber esbozado un primer modelo de las clases que serán necesarias para el almacenamiento de los datos para esta aplicación, se necesita cumplir con tres pilares imprescindibles para el desarrollo de esta herramienta. Estos tres apartados a diseñar son la **Base de Datos** que almacenará la información de nuestra herramienta, el **Modelo de Clases final**, el cual será el diseño final interno de este proyecto y la **Infraestructura**, que será la encargada de mantener la herramienta siempre disponible.

6.1.1. Base de Datos

Como primer paso tras haber analizado las necesidades de la herramienta mediante sus casos de uso y los diferentes diagramas de acciones que debe realizar la aplicación, se hace preciso la creación de un lugar donde toda esa información resida para ser usada por la herramienta.

Para ello, en este proyecto se ha optado por el uso de una base de datos de tipo `mySQL`, donde la información puede ser introducida de manera sencilla y rápida desde la propia herramienta.

Dado que desde el inicio de este proyecto se fijó que la herramienta debería ser accesible en todo momento, pese al trabajo simultáneo del desarrollador, la herramienta deberá utilizar una base de datos a nivel local para el desarrollo de la aplicación, creada por el autor del presente documento, así como, encontrarse disponible más allá del ámbito local mediante un servicio web que será acordado con el cliente y que no entra dentro del ámbito del desarrollo de éste proyecto.

Esta base de datos local, será creada de manera básica desde el software SQLDeveloper mediante la ejecución de los archivos SQL. Estas acciones darán como resultado la introducción de los modelos citados en nuestra base de datos local, donde podremos trabajar con la información que se introduzca en la herramienta desde el propio navegador mediante el uso del cliente o mediante sentencias SQL.

En nuestro caso la base de datos local será alojada en la dirección `http://127.0.0.1:8000` y su usuario y contraseña será elegido por el autor de este documento, pues será el encargado de su administración.

En la siguiente figura se puede apreciar la BD ya creada, aunque sin datos en los mismos, pues aún se precisa de refinar el modelo de clases definitivo para la herramienta desarrollada en este proyecto.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
AUDITURIA	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	32 KB	-
CONTROL_SALIDA	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	96 KB	-
HERMANO	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8_general_ci	88 KB	-
OCUPACION	Examinar Estructura Buscar Insertar Vaciar Eliminar	64	InnoDB	utf8_general_ci	32 KB	-
PASO	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	32 KB	-
PERSONA	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8_general_ci	48 KB	-
PROCESION	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_general_ci	48 KB	-
RECIBO	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8_general_ci	64 KB	-
sessions	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	16 KB	-
TIPO_PAGO	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_general_ci	32 KB	-
TIPO_RECIBO	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8_general_ci	16 KB	-
TUNICA	Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8_general_ci	48 KB	-
Usuario	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8_general_ci	32 KB	-
13 tablas	Número de filas	186	InnoDB	utf8_general_ci	576 KB	0 B

Figura 6.1: Base de Datos local.

6.1.2. Modelo de clases

Tras seleccionar la base de datos que será usada para el almacenamiento de la información, se utiliza el modelo de clases desarrollado en el anterior capítulo en busca de fallas en el sistema pensado, o si es necesario la inclusión de nuevos modelos para que toda la información quede bien registrada en la herramienta conforme a las necesidades del desarrollador y el usuario final.

A medida que el proyecto iba incrementando de forma gradual en complejidad y funcionalidades en cada fase del desarrollo, se hizo notar que el modelo de clases inicial no recogía ciertos aspectos para el modelo relacionado con el ámbito Hermano-Procesion, por lo que fue necesario realizar nuevos modelos de clases e ir adaptándolos para incluir mejoras en la herramienta.

Por ello, se modificó el diagrama de clases creando una clase intermedia.

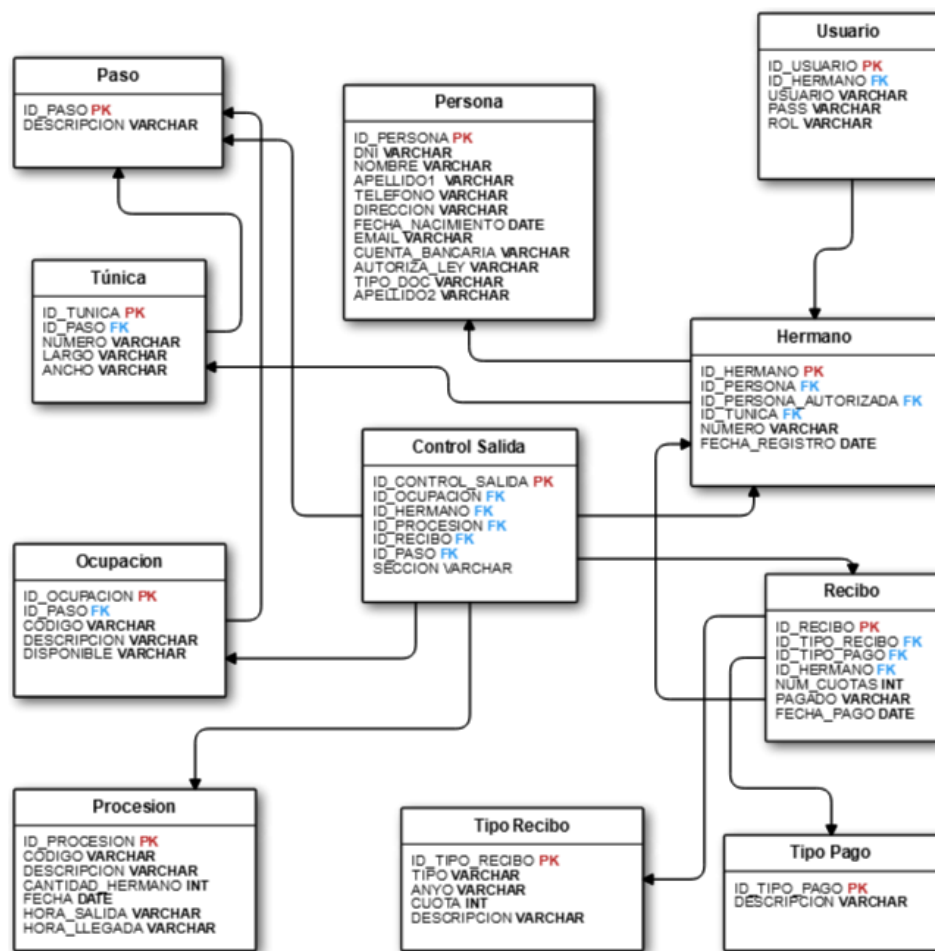


Figura 6.2: Diagrama Final

6.1.3. Infraestructura

Una vez fijado el modelo de clases definitivo que deberá utilizar la herramienta se hace necesario cumplir con la premisa de que la herramienta fuese accesible en cualquier lugar y momento, por lo que se acordó un desarrollo pensado para que la aplicación se pudiese alojar en hosting que tuviesen las tecnologías con las que trabajamos.

Este servicio web con la herramienta en su interior poseerá su propia BD, ajena a la base de datos de pruebas para el desarrollador que se mostró anteriormente, pero con su misma estructura y funcionamiento, donde sus características son:

- Base de Datos: MySQL
- Servidor: Apache/Weblogic
- Sistema de Versiones: Git

Una vez la herramienta esté finalizada será utilizada por el cliente desde un servidor externo contrata en una empresa externa donde deberá adjudicarse un dominio y donde se cumplirá con la necesidad de su uso continuado siendo esta última parte no perteneciente al ámbito de éste proyecto.

6.1.4. Vistas y formularios

La herramienta se desarrolla en base a diversas vistas para los distintos usuarios de la misma, tal y como se indicó en el capítulo de descripción, donde la herramienta debe ser capaz de realizar diversas acciones mediante dichas vistas y el uso de sus formularios asociados.

A continuación se describen diferentes tipos de formularios y vistas que puede hacer uso la herramienta para realizar las acciones necesarias para el buen funcionamiento de la misma. Se utilizará como ejemplo visual el módulo de 'Alta de Control de Salida' pues es de los módulos más completos dentro de la herramienta, así como, cualquier otra vista especial que contenga la aplicación para su completa comprensión.

6.1.4.1. Vista General

En la vista general obtenemos una presentación de la herramienta. Dicha vista tiene su mayor peso en un login de usuario para la muestra de división del proyecto en roles y su diferentes funcionalidades.

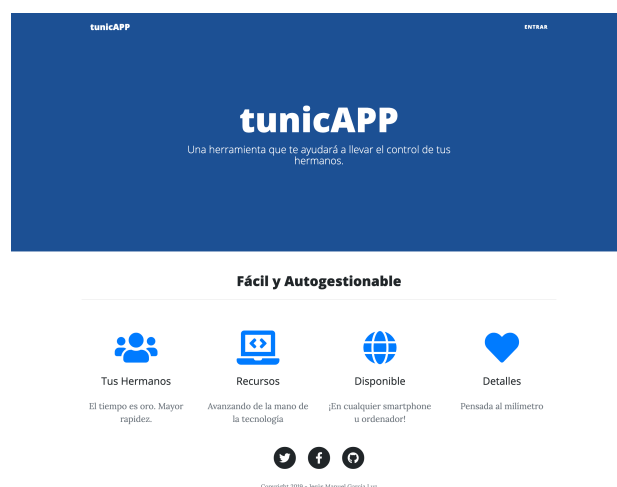


Figura 6.3: Inicio

6.1.4.2. Vista principal del Hermano

En la vista del Hermano, el desarrollador acorde con el cliente ha decidido que herramientas poseerá el usuario Hermano para el uso de la herramienta.

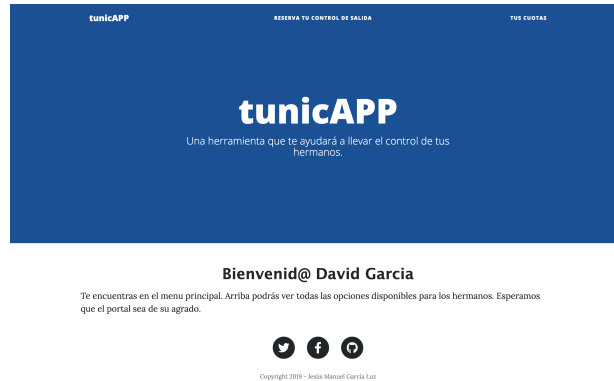


Figura 6.4: Panel del Hermano

6.1.4.3. Vista principal del Administrador

En la vista del Administrador, el desarrollador acorde con el cliente ha decidido que herramientas poseerá el usuario Administrador para el uso de la herramienta.

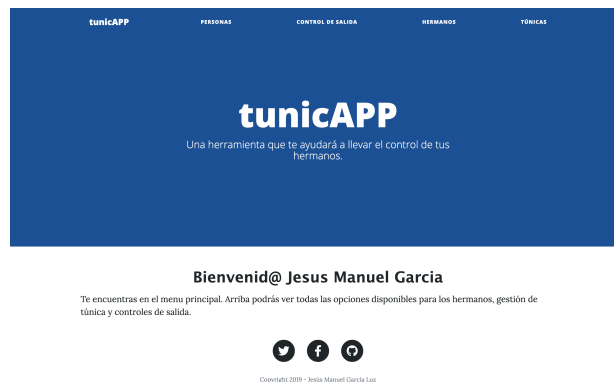


Figura 6.5: Panel del Administrador

6.1.4.4. Alta de Control de Salida

Haciendo click en el menú superior Controles de Salida y posteriormente a Nuevo Control de Salida se le muestra al usuario un formulario que debe rellenar para el alta del Control de Salida asociado a un Hermano ya existente en la aplicación.

Estos formularios de alta, variarán en función de las necesidades de cada Hermano, ya que se debe rellenar información como titular al que quiere acompañar, insignia que quiere llevar y método de pago seleccionado. Cada información es personal para cada Hermano.

The screenshot shows the 'tunicAPP' interface with a dark blue header. The main content area is white and titled 'Alta Control de Salida'. It contains several sections: 1. 'Panel de Búsqueda' with a DNI/NIE input field containing '76647220L' and 'Buscar'/'Limpiar' buttons. 2. 'Datos de Jesus Manuel Garcia Luz:' with fields for DNI/NIE, Número de Documento, Nombre, Apellidos, and a text field for 'Garcia Luz'. 3. 'Control de Salida' with 'Tipo de Recibo (*)' set to 'Salida Procesional 2019', 'Cuantía' set to '20', and a 'Pagado' dropdown set to 'No'. 4. 'Túnica' with 'Titular (*)' and 'Portador (*)' dropdowns set to 'Seleccione', and a 'Túnica (*)' dropdown set to 'Seleccione'. At the bottom is a 'Dar de Alta el Control' button, social media icons, and a copyright notice: 'Copyright 2019 - Jesus Manuel Garcia Luz'.

Figura 6.6: Alta de Control de Salida

La siguiente figura muestra como, tras introducir un registro erróneo del formulario, la aplicación genera un mensaje para que el usuario administrador del mismo pueda corregir la información y volver a realizar el proceso de alta.

The screenshot displays the tunicAPP interface. At the top, the logo 'tunicAPP' is shown with the tagline 'Una herramienta que te ayudará a llevar el control de tus hermanos.' To the right, a yellow error box contains the text: 'Error en el alta', 'El hermano tiene una cuota de salida', and 'Hermano sin pagar. Debe pagarla antes de crear una cuota de Salida.' Below this, the section 'Alta Control de Salida' is visible. Underneath, there is a 'Panel de Búsqueda' which includes a text input field for 'DNI/NIE (*)' containing the value '76647220L', and two buttons labeled 'Buscar' and 'Limpiar'.

Figura 6.7: Validación de Alta Hermano

Capítulo 7

Implementación

En este capítulo se expondrá la forma de trabajo utilizada para el desarrollo de la herramienta mediante la programación Java, así como, la implementación del aspecto visual de la herramienta y el código fuente de acciones de interés dentro de esta.

7.1. Programación en Java

Para este desarrollo por conocimiento del creador del mismo se ha utilizado el lenguaje Java. Por complejidad del proyecto, el creador del mismo ha decidido utilizar varias tecnologías como son JSF, JPA y PrimeFaces.

Dichas tecnologías sumadas al lenguaje Java, ofrecen la posibilidad de implementar un MVC en nuestro desarrollo.

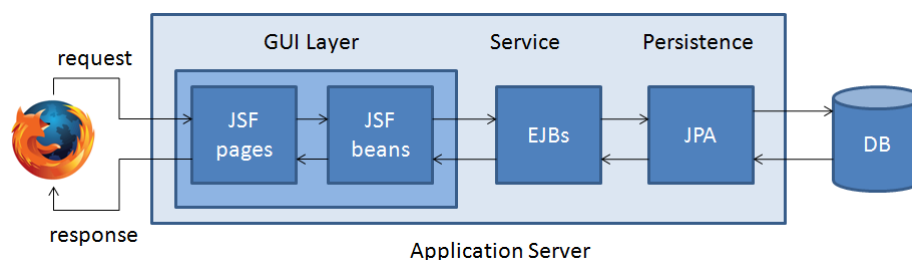


Figura 7.1: Arquitectura Java

En nuestro caso, hemos usado la versión JSF 2.0, la versión JPA 2.1 y la versión 6.2 en PrimeFaces para el desarrollo de nuestro proyecto.

7.1.1. JPA

El uso de JPA facilita al desarrollador la implementación de su base de datos en Java, viéndola como una creación de objetos que, posteriormente por el entity-manager, persistirá en nuestra base de datos pudiendo así manejar, clonar, y borrar objetos en Java recogidos de la base de datos pero que, mientras no pasen por nuestro entity-manager, no se verán reflejados en nuestra base de datos.

La programación en JPA es muy conocida en el mercado laboral y la inmensa mayoría de empresas que implementan estas tecnologías y tienen proyectos Java la utilizan. Es por ello que existe un patrón de diseño sobre JPA, para que todos los equipos de desarrolladores existentes conozcan las bases fundamentales y no se bifurque la forma de trabajar en los desarrollos.

Aunque existen innumerables ventajas de usar JPA en el desarrollo de nuestras aplicaciones Java a continuación voy a mencionar algunas:

- La carga de interactuar con la base de datos se reduce significativamente mediante el uso de JPA
- La programación del usuario se vuelve fácil al ocultar el mapeo directo desde un modelo Java.
- El costo de crear el archivo de definición (xml) se reduce mediante el uso de anotaciones

Una de las mayores ventajas expuestas en el anterior apartado son el uso de anotaciones. Las mas importantes son las siguientes:

- @Entity Usada para determinar los objetos Java que van a representar a objetos de nuestra base de datos. Para ser Entity, el patrón de diseño recomienda que se cumpla:
 - Proporcionar un constructor por defecto (ya sea de forma implícita o explícita).
 - Ser una clase de primer nivel (no interna).
 - No ser final.
 - Implementar la interface java.io.Serializable si va a ser accedida remotamente.
- Todas las entidades tienen que poseer una identidad que las diferencie del resto, por lo que deben contener una propiedad marcada con la anotación @Id.
- La identidad de una entidad va a ser gestionada por el proveedor de persistencia, así que será dicho proveedor quien le asigne un valor la primera vez que almacene la entidad en la base de datos. Para tal efecto, le añadimos a la propiedad de identidad la anotación @GeneratedValue.
- La anotación @Table nos permite configurar el nombre de la tabla donde queremos almacenar la entidad mediante el atributo name.
- Con la anotación @Column podemos configurar el nombre de la columna donde se almacenará una propiedad.
- Al recuperar un objeto de base de datos podemos decidir si su lectura se hace en el momento de consultar la base de datos o que venga ya recogido desde la creación del mismo. Decidir bien este tipo de consulta nos da la posibilidad de recoger datos muy pesados o no, para poder así agilizar el comportamiento del software. Estas propiedades son:
 - @Basic(fetch = FetchType.EAGER)
 - @Basic(fetch = FetchType.LAZY)
- JPA puede mapear los tipos enumerados (enum) mediante la anotación @Enumerated.

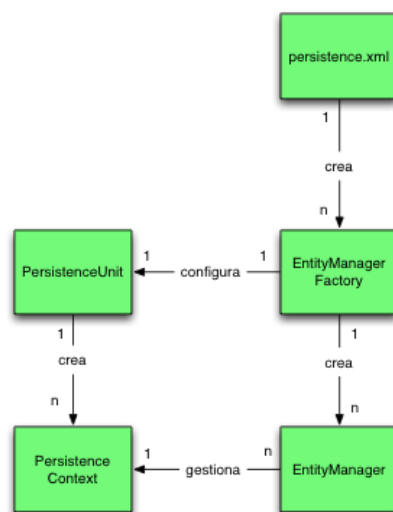


Figura 7.2: Persistencia de JPA

7.1.2. JSF

JSF es un framework MVC (Modelo-Vista-Controlador) basado en el API de Servlets que proporciona un conjunto de componentes en forma de etiquetas definidas en páginas XHTML mediante el framework Facelets. Facelets se define en la especificación 2 de JSF como un elemento fundamental de JSF que proporciona características de plantillas y de creación de componentes compuestos. Antes de la especificación actual se utilizaba JSP para componer las páginas JSF.

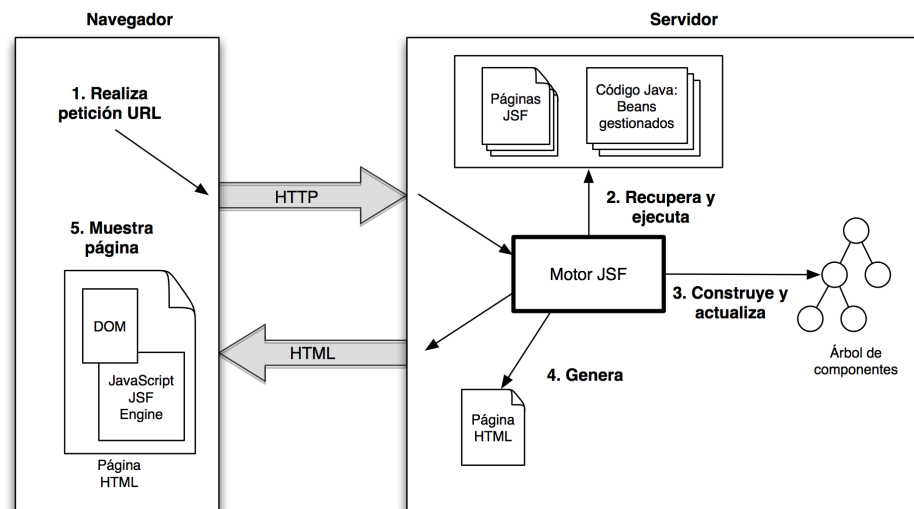


Figura 7.3: Diagrama básico de JSF

Entrando un poco más en detalle, JSF proporciona las siguientes características destacables:

- Definición de las interfaces de usuario mediante vistas que agrupan componentes gráficos.
- Conexión de los componentes gráficos con los datos de la aplicación mediante los denominados beans gestionados.
- Conversión de datos y validación automática de la entrada del usuario.
- Navegación entre vistas.
- A partir de la especificación 2.0 un modelo estándar de comunicación Ajax entre la vista y el servidor.

JSF además de proporcionarnos una API de componentes gráficos, también nos permite gestionar y distribuir la información por nuestros Beans. Un bean es una clase con un conjunto de atributos (denominados propiedades) y métodos getters y setters que devuelven y actualizan sus valores. Las propiedades del bean se pueden leer y escribir desde las páginas JSF. Dichos Beans, a partir de la versión 2.0 de JSF, pueden usar las siguientes etiquetas:

- **Petición:** Se define la anotación `@RequestScoped` en la clase. El bean se asocia a una petición HTTP. Cada nueva petición (cuando desde el navegador se abre una página por primera vez una página o se recarga) crea un nuevo bean y lo asocia con la página. Dada su corta vida, se recomienda usar este ámbito para el paso de mensajes (bien sea de error o de estatus), o para cualquier otro tipo de dato que no sea necesario propagar a lo largo de la aplicación.
- **Sesión:** Se define con la anotación `@SessionScoped` en la clase. Las sesiones se definen internamente con el API de Servlets. Una sesión está asociada con una visita desde un navegador. Cuando se visita la página por primera vez se inicia la sesión. Cualquier página que se abra dentro del mismo navegador comparte la sesión. La sesión mantiene el estado de los elementos de nuestra aplicación a lo largo de las distintas peticiones.

- **Aplicación:** Se define con la anotación `@ApplicationScoped`. Los beans con este ámbito viven asociados a la aplicación. Definen singletons que se crean e inicializa sólo una vez, al comienzo de la aplicación. Se suelen utilizar para guardar características comunes compartidas y utilizadas por el resto de beans de la aplicación.
- **Vista:** Se define con la anotación `@ViewScoped` en la clase a partir de JSF 2.0. Un bean en este ámbito persistirá mientras se repinte la misma página (vista = página JSF), al navegar a otra página, el bean sale del ámbito. Es bastante útil para aplicaciones que usen Ajax en parte de sus páginas.
- **Custom:** Se define con la anotación `@CustomScoped`. Un ámbito al fin y al cabo no es más que un mapa que enlaza nombres y objetos. Lo que distingue un ámbito de otro es el tiempo de vida de ese mapa. Los tiempos de vida de los ámbitos estándar de JSF (sesión, aplicación, vista y petición) son gestionados por la implementación de JSF. En JSF 2.0 podemos crear ámbitos personalizados, que son mapas cuyo ciclo de vida gestionamos nosotros. Para incluirlo en ese mapa, usaremos la anotación `@CustomScoped`. Nuestra aplicación será la responsable de eliminar elementos de ese mapa. Es una configuración de JSF que no incluiremos en nuestra aplicación ya que no es necesaria tanta complejidad para nuestro análisis y desarrollo.
- **Conversación:** Se define con la etiqueta `@ConversationScoped`. Provee de persistencia de datos hasta que se llega a un objetivo específico, sin necesidad de mantenerlo durante toda la sesión. Está ligado a una ventana o pestaña concreta del navegador. Así, una sesión puede mantener varias conversaciones en distintas páginas. Es una característica propia de CDI, no de JSF.

JSF se ejecuta sobre la tecnología de Servlets y no requiere ningún servicio adicional, por lo que para ejecutar aplicaciones JSF sólo necesitamos un contenedor de servlets tipo Tomcat o Jetty. Para nuestro proyecto, hemos usado Tomcat.

7.1.3. PrimeFaces

Para nuestro proyecto y en apoyo a JSF, hemos usado la librería de PrimeFaces. PrimeFaces es uno de los frameworks web con mayor aceptación hoy día y que muchos programadores web han puesto en práctica con resultados favorables. Destacar que es una librería de componentes visuales Open Source para JSF.

Al igual que JSF, PrimeFaces nos ofrece componentes gráficos para usarlos en nuestros XHTML y representar así, la información manejada en nuestros Beans de Java.



Figura 7.4: Logo de PrimeFaces

7.2. Interfaz de la herramienta

Una vez comprendido el funcionamiento interno de Java, JSF, PrimeFaces, Bootstrap y JPA para el desarrollo de la herramienta, en este apartado nos centraremos en la interfaz de la aplicación y la forma en que esta muestra la información a los usuarios de la misma.

Hemos tomado como referencia un patrón de diseño CSS, ya que el conocimiento en el apartado front del creador del proyecto es escaso. Un patrón de diseño es diferente a una plantilla, por lo

que cabe destacar que todo el CSS desarrollado es íntegramente de cero y que solo se ha apoyado en la utilización de los componentes gráficos de JSF y PrimeFaces.

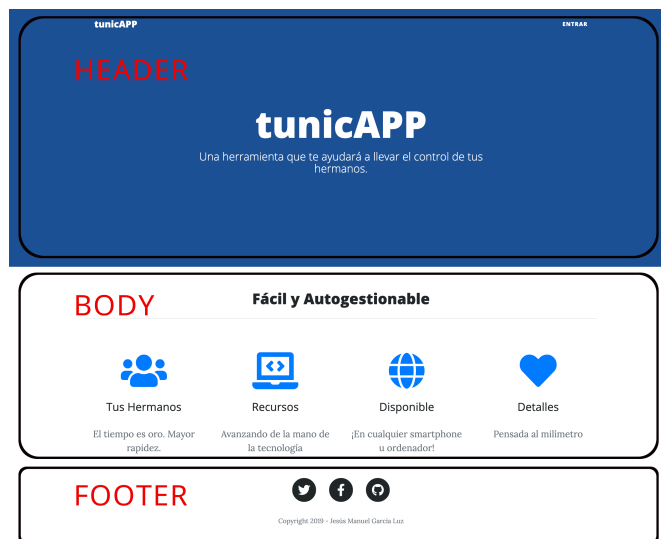


Figura 7.5: Estructura de la Interfaz

```

1  <!DOCTYPE html>
2  <html lang="en" xmlns="http://www.w3.org/1999/xhtml"
3  xmlns:p="http://primefaces.org/ui"
4  xmlns:f="http://java.sun.com/jsf/core"
5  xmlns:h="http://java.sun.com/jsf/html"
6  xmlns:ui="http://java.sun.com/jsf/facelets">
7
8  <h:head>
9
10
11
12  <meta charset="utf-8" />
13
14  <title>tunicAPP</title>
15
16  <link href="css/bootstrap-datetimepicker.min.css" rel="stylesheet"></link>
17  <script src="js/bootstrap-datetimepicker.min.js"></script>
18  <!-- Bootstrap core CSS -->
19  <link href="resources/vendor/bootstrap/css/bootstrap.min.css"
20  rel="stylesheet" />
21  <link href="resources/css/estilos.css" rel="stylesheet" />
22
23  <!-- Custom fonts for this template -->
24  <link href="resources/vendor/fontawesome-free/css/all.min.css"
25  rel="stylesheet" type="text/css" />
26  <link
27  href='https://fonts.googleapis.com/css?family=Lora:400,700,400italic,700italic'
28  rel='stylesheet' type='text/css' />
29  <link
30  href='https://fonts.googleapis.com/css?family=Open+Sans:300italic,400italic,600
    italic,700italic,800italic,400,300,600,700,800'
31  rel='stylesheet' type='text/css' />
32
33  <!-- Custom styles for this template -->
34  <link href="resources/css/clean-blog.min.css" rel="stylesheet" />
35
36  </h:head>
37
38  <h:body>
39
40  <!-- Navigation -->
41  <nav class="navbar navbar-expand-lg navbar-light fixed-top"
42  id="mainNav">
43  <div class="container">
44  <a class="navbar-brand" href="index.xhtml">tunicAPP</a>
45  <button class="navbar-toggler navbar-toggler-right" type="button"
46  data-toggle="collapse" data-target="#navbarResponsive"
47  aria-controls="navbarResponsive" aria-expanded="false"
48  aria-label="Toggle navigation">
49  Menu <i class="fas fa-bars"></i>
50  </button>
51  <div class="collapse navbar-collapse" id="navbarResponsive">
52  <ul class="navbar-nav ml-auto">
53  <li class="nav-item dropdown"><a
54  class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
55  role="button" data-toggle="dropdown" aria-haspopup="true"
56  aria-expanded="false"> Personas </a>
57  <div class="dropdown-menu" aria-labelledby="navbarDropdown">
58  <a class="dropdown-item" href="personas_list.xhtml">Listado</a> <a
59  class="dropdown-item" href="personas_add.xhtml">Añadir</a> <a
60  class="dropdown-item" href="personas_edit.xhtml">Editar</a> <a
61  class="dropdown-item" href="personas_delete.xhtml">Eliminar</a>
62  </div></li>
63  <li class="nav-item dropdown"><a
64  class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
65  role="button" data-toggle="dropdown" aria-haspopup="true"
66  aria-expanded="false"> Control de Salida </a>
67  <div class="dropdown-menu" aria-labelledby="navbarDropdown">
68  <a class="dropdown-item" href="control_list.xhtml">Listado</a> <a
69  class="dropdown-item" href="control_add.xhtml">Añadir</a> <a

```

```

70 class="dropdown-item" href="control_edit.xhtml">Editar</a> <a
71 class="dropdown-item" href="control_delete.xhtml">Eliminar</a>
72 </div></li>
73 <li class="nav-item dropdown"><a
74 class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
75 role="button" data-toggle="dropdown" aria-haspopup="true"
76 aria-expanded="false"> Hermanos </a>
77 <div class="dropdown-menu" aria-labelledby="navbarDropdown">
78 <a class="dropdown-item" href="hermanos_list.xhtml">Listado</a> <a
79 class="dropdown-item" href="hermanos_add.xhtml">Añadir</a><a
80 class="dropdown-item" href="hermanos_edit.xhtml">Editar</a> <a
81 class="dropdown-item" href="hermanos_delete.xhtml">Eliminar</a>
82 </div></li>
83 <li class="nav-item dropdown"><a
84 class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
85 role="button" data-toggle="dropdown" aria-haspopup="true"
86 aria-expanded="false"> Túnicas </a>
87 <div class="dropdown-menu" aria-labelledby="navbarDropdown">
88 <a class="dropdown-item" href="tunicas_list.xhtml">Listado</a> <a
89 class="dropdown-item" href="tunicas_add.xhtml">Añadir</a><a
90 class="dropdown-item" href="tunicas_edit.xhtml">Editar</a> <a
91 class="dropdown-item" href="tunicas_delete.xhtml">Eliminar</a>
92 </div></li>
93 </ul>
94 </div>
95 </div>
96 </nav>
97
98 <!-- Page Header -->
99 <header class="masthead" style="background-color: #007bff">
100 <div class="overlay"></div>
101 <div class="container">
102 <div class="row">
103 <div class="col-lg-8 col-md-10 mx-auto">
104 <div class="site-heading">
105 <h1>tunicAPP</h1>
106 <span class="subheading">Una herramienta que te ayudará a
107 llevar el control de tus hermanos.</span>
108 </div>
109 </div>
110 </div>
111 </div>
112 </header>
113
114
115
116 <!-- Services Section -->
117 <section class="page-section" id="services">
118 <h:form id="form">
119 <div class="container">
120 <h2 class="text-center mt-0">Alta de Hermano</h2>
121 <hr class="divider my-4"></hr>
122
123 <p:growl id="growl" showDetail="true" life="3000">
124 <p:autoUpdate />
125 </p:growl>
126 <h:panelGrid id="panelBusqueda">
127 <p:fieldset id="busqueda" legend="Panel de Búsqueda">
128 <div class="form-row">
129 <div class="col">
130 <label for="tipoDni">DNI/NIE (*)</label>
131 <div class="col">
132 <h:inputText placeholder="Número de documento" id="dniBusqueda"
133 required="true" label="Número de Documento" update="@form"
134 value="#{hermanosAddBean.hermano.persona.dni}" maxLength="9" />
135 </div>
136 </div>
137 </div>
138 <p:spacer height="10"></p:spacer>
139 <div class="form-row">

```

```

140 <div class="col">
141 <p:commandButton update="@form" value="Buscar" id="btnBuscar"
142 action="#{hermanosAddBean.buscarPersona()}" />
143 <p:commandButton update="@form" value="Limpiar" id="btnLimpiar"
144 action="#{hermanosAddBean.init()}" />
145 </div>
146 </div>
147 </p:fieldset>
148 </h:panelGrid>
149 <p:spacer height="50"></p:spacer>
150 <h:panelGrid id="panelDatos">
151 <p:fieldset id="datos"
152 rendered="#{hermanosAddBean.hermano.persona.nombre ne null}"
153 legend="Datos de #{hermanosAddBean.hermano.persona.nombre} #{
    hermanosAddBean.hermano.persona.apellido1} #{
    hermanosAddBean.hermano.persona.apellido2}:" >
154 <div class="form-row">
155 <div class="col">
156 <label for="tipoDni">DNI/NIE</label>
157 <div class="col">
158 <h:inputText placeholder="Número de documento" id="dni"
159 required="true" label="Número de Documento" readonly="true"
160 value="#{hermanosAddBean.hermano.persona.tipoDoc}"
161 maxlength="9" />
162 </div>
163 </div>
164 <div class="col">
165 <label for="dni">Número de Documento</label>
166 <div class="col">
167 <h:inputText placeholder="Número de documento" id="PersonaDni"
168 required="true" label="Número de Documento" readonly="true"
169 value="#{hermanosAddBean.hermano.persona.dni}" maxlength="9" />
170 </div>
171 </div>
172 </div>
173 <p:spacer height="5"></p:spacer>
174 <div class="form-row">
175 <div class="col">
176 <label for="nombre">Nombre</label>
177 <div class="col">
178 <h:inputText placeholder="Nombre" id="nombre" required="true"
179 readonly="true" label="Nombre"
180 value="#{hermanosAddBean.hermano.persona.nombre}">
181
182 </h:inputText>
183 </div>
184 </div>
185 <div class="col">
186 <label for="primerApellido">Primer Apellido</label>
187 <div class="col">
188 <h:inputText placeholder="Primer Apellido" id="primerApellido"
189 required="true" label="Primer Apellido" readonly="true"
190 value="#{hermanosAddBean.hermano.persona.apellido1}" />
191 </div>
192 </div>
193 </div>
194 <p:spacer height="5"></p:spacer>
195 <div class="form-row">
196 <div class="col">
197 <label for="segundoApellido">Segundo Apellido</label>
198 <div class="col">
199 <h:inputText placeholder="Segundo Apellido"
200 id="segundoApellido" readonly="true"
201 value="#{hermanosAddBean.hermano.persona.apellido2}" />
202 </div>
203 </div>
204 <div class="col">
205 <label for="telefono">Teléfono</label>
206 <div class="col">
207 <h:inputText placeholder="Teléfono" id="telefono"

```

```

208 required="true" label="Teléfono" readonly="true"
209 value="#{hermanosAddBean.hermano.persona.telefono}"
210 maxLength="9" />
211 </div>
212 </div>
213 <div class="col">
214 <label for="email">Email</label>
215 <div class="col">
216 <h:inputText placeholder="email" id="email" readonly="true"
217 value="#{hermanosAddBean.hermano.persona.email}" />
218 </div>
219 </div>
220 </div>
221 <p:spacer height="5"></p:spacer>
222 <div class="form-row">
223 <div class="col">
224 <label for="fecha">Fecha de Nacimiento</label>
225 <div class="col">
226 <span class="ui-float-label"> <p:inputText
227 required="true" label="Fecha de Nacimiento" readonly="true"
228 placeholder="dia/mes/año" id="fecha"
229 converter="dateToStringConverter"
230 value="#{hermanosAddBean.hermano.persona.fechaNacimiento}" />
231 </span>
232 </div>
233 </div>
234 <div class="col">
235 <h:panelGroup id="panelAutorizado"
236 rendered="#{hermanosAddBean.comprobarEdad()}">
237 <label for="personaAut">DNI Autorizado</label>
238 <h:inputText placeholder="DNI Autorizado" id="dniAutorizado"
239 required="true" label="DNI Persona Autorizada"
240 value="#{hermanosAddBean.hermano.personaAutorizada.dni}"
241 maxLength="9" />
242 <p:commandButton update="form datos" value="Buscar"
243 id="btnBuscarAutorizado"
244 action="#{hermanosAddBean.buscarPersonaAutorizada()}" />
245 </h:panelGroup>
246 </div>
247 </div>
248 <p:spacer height="5"></p:spacer>
249 <div class="form-row">
250 <div class="col">
251 <label for="domicilio">Domicilio</label>
252 <div class="col">
253 <h:inputText style="width:750px;" placeholder="Domicilio"
254 readonly="true" required="true" label="Domicilio"
255 id="domicilio"
256 value="#{hermanosAddBean.hermano.persona.direccion}" />
257 </div>
258 </div>
259 </div>
260 <p:spacer height="5"></p:spacer>
261 <div class="form-row">
262 <div class="col">
263 <label for="autoriza">Autoriza LOPD</label>
264 <div class="col">
265 <h:inputText style="width:750px;" placeholder="Autoriza"
266 readonly="true"
267 value="#{hermanosAddBean.hermano.persona.autorizaLey}"
268 required="true" label="Autoriza" id="autoriza"
269 converter="booleanStringSNConverter" />
270 </div>
271 </div>
272 </div>
273 </p:fieldset>
274 </h:panelGrid>
275 <p:spacer height="10"></p:spacer>
276 <h:panelGrid id="panelRecibo">
277 <p:fieldset id="reciboHermano" legend="Cuota de Hermano"

```

```

278 rendered="{#{hermanosAddBean.hermano.persona.nombre ne null}}">
279 <div class="form-row">
280 <div class="col">
281 <label for="tipoRecibo">Tipo de Recibo (*)</label>
282 <div class="col">
283 <h:selectOneMenu id="recibo" label="Tipo de Recibo"
284 required="true" value="{#{hermanosAddBean.recibo.tipoRecibo}}"
285 converter="{#{hermanosAddBean.getTiposReciboConverter(hermanosAddBean.listTipoRecibo)
    }}">
286 <f:selectItems value="{#{hermanosAddBean.listTipoRecibo}}"
287 var="recibo" itemValue="{#{recibo}}"
288 itemLabel="{#{recibo.descripcion}}" />
289 </h:selectOneMenu>
290 </div>
291 </div>
292 <div class="col">
293 <label for="numCuota">Número de cuotas (*)</label>
294 <div class="col">
295 <h:inputText placeholder="Número de cuotas" id="numCuotas"
296 value="{#{hermanosAddBean.recibo.numeroCuotas}}" />
297 </div>
298 </div>
299 </div>
300 <div class="form-row">
301 <div class="col">
302 <label for="tipoRecibo">Tipo de Pago (*)</label>
303 <div class="col">
304 <h:selectOneMenu id="tipoPago" label="Tipo de Pago"
305 required="true" value="{#{hermanosAddBean.recibo.tipoPago}}"
306 converter="{#{hermanosAddBean.getTiposPagoConverter(hermanosAddBean.listTipoPago)}">
307 <f:selectItems value="{#{hermanosAddBean.listTipoPago}}"
308 var="tipoPago" itemValue="{#{tipoPago}}"
309 itemLabel="{#{tipoPago.descripcion}}" />
310 <f:ajax listener="{#{hermanosAddBean.listener}" render="@form"></f:ajax>
311 </h:selectOneMenu>
312 </div>
313 </div>
314 <p:panelGrid id="panelCuentaBancaria"
315 rendered="{#{hermanosAddBean.recibo.tipoPago.descripcion eq 'Transferencia Bancaria'}"
    ">
316 <div class="col">
317 <label for="banco">Cuenta Bancaria</label>
318 <div class="col">
319 <p:inputText style="width:750px;"
320 placeholder="Cuenta Bancaria" id="banco"
321 value="{#{hermanosAddBean.hermano.persona.cuentaBancaria}">
322 </p:inputText>
323 </div>
324 </div>
325 </p:panelGrid>
326 </div>
327 </p:fieldset>
328 </h:panelGrid>
329 <p:spacer height="10"></p:spacer>
330 <div class="form-row">
331 <p:commandButton update="@form" value="Dar de Alta"
332 rendered="{#{hermanosAddBean.hermano.persona.nombre ne null}"
333 action="{#{hermanosAddBean.altaHermano()}" />
334 </div>
335 </div>
336 </h:form>
337 </section>
338
339
340
341
342 <!-- Footer -->
343 <footer>
344 <div class="container">
345 <div class="row">

```

```

346 <div class="col-lg-8 col-md-10 mx-auto">
347 <ul class="list-inline text-center">
348 <li class="list-inline-item"><a
349 href="https://twitter.com/cofradiaperdon?lang=es"> <span
350 class="fa-stack fa-lg"> <i
351 class="fas fa-circle fa-stack-2x"></i> <i
352 class="fab fa-twitter fa-stack-1x fa-inverse"></i>
353 </span>
354 </a></li>
355 <li class="list-inline-item"><a
356 href="https://www.facebook.com/perdonyrosario/"> <span
357 class="fa-stack fa-lg"> <i
358 class="fas fa-circle fa-stack-2x"></i> <i
359 class="fab fa-facebook-f fa-stack-1x fa-inverse"></i>
360 </span>
361 </a></li>
362 <li class="list-inline-item"><a
363 href="https://github.com/YiseKai/TFG"> <span
364 class="fa-stack fa-lg"> <i
365 class="fas fa-circle fa-stack-2x"></i> <i
366 class="fab fa-github fa-stack-1x fa-inverse"></i>
367 </span>
368 </a></li>
369 </ul>
370 <p class="copyright text-muted">Copyright 2019 - Jesús Manuel
371 García Luz</p>
372 </div>
373 </div>
374 </div>
375 </footer>
376
377 <!-- Bootstrap core JavaScript -->
378 <script src="resources/vendor/jquery.js"></script>
379 <script src="resources/vendor/jquery/jquery.min.js"></script>
380 <script src="resources/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
381
382 <!-- Custom scripts for this template -->
383 <script src="resources/js/clean-blog.min.js"></script>
384 </h:body>
385
386 </html>

```

Código 7.1: Arquitectura Front del aplicativo

Este diseño de la interfaz dota a la aplicación de un toque moderno basado en Bootstrap que se integra de forma natural con los componentes gráficos de JSF y PrimeFaces. Sobre dicha base se orquesta toda la arquitectura front de nuestro proyecto y se muestra en la siguiente figura:

The screenshot displays the tunicAPP interface. At the top, a blue header contains the text 'tunicAPP' and 'Una herramienta que te ayudará a llevar el control de tus hermanos.' Below this, a section titled 'Alta Control de Salida' contains a form for 'Datos de David García Luz:'. The form includes fields for 'DNI/NIE' (DNI), 'Número de Documento' (03788967Q), 'Nombre' (David), and 'Apellidos' (García Luz). Below this is a 'Control de Salida' section with fields for 'Tipo de Recibo (*)' (Salida Procesional 2019), 'Cuenta' (20), 'Pagado' (No), and 'Tipo de Pago (*)' (Metálico). A 'Comprobar' button is next to the 'Pagado' field. At the bottom, there is a 'Túnica' section with 'Titular (*)' and 'Túnica (*)' dropdown menus, both set to 'Seleccione'. A 'Dar de Alta el Control' button is at the very bottom. Social media icons for Twitter, Facebook, and YouTube are also present.

Figura 7.6: Alta de Hermano.

7.3. Funciones de interés sobre los componentes web

Debido a la forma de trabajar con JSF y PrimeFaces, cada componente posee una serie de propiedades específicas del mismo para poder hacer uso de ellos. Entre todas las propiedades diferentes, obtenemos las mas usadas del componente `commandButton`:

- **Id** Es el ID del componente para el formulario HTML.
- **Value** Es el texto que se mostrará junto al botón en la web.
- **Action** Función que realiza el botón al realizar una orden. Puede conllevar una navegabilidad.
- **ActionListener** Función que realiza el botón al realizar una orden. No puede llevar navegabilidad.
- **Disable** Propiedad que indica si el componente se encuentra activado o desactivado.
- **Rendered** Propiedad que indica si el componente se muestra en pantalla o no.
- **Update** Tras realizar una acción sobre el botón, se ordena actualizar los componentes que se indiquen dentro de la propiedad.
- **StyleClass** Funcionalidad que permite asignar una clase CSS al botón.
- **Style** Propiedad que complementa el `styleClass` del componente.

7.3.1. Verificación de campos

Para la verificación de los campos que posee un formulario dentro de nuestra herramienta, existen dos formas diferentes. Validación por cliente, que sería una validación mediante Ajax dónde la validación nunca llega a hacer una comunicación con el servidor por lo que la carga es ligera y validación por servidor, que se suele hacer para formularios completos.

En el caso de una validación por cliente se realiza mediante los componentes de PrimeFaces/JSF:

```
1 <h:inputText placeholder="Número de documento" id="dni"
2 required="true" label="Número de Documento" readonly="true"
3 value="#{controlSalidaAddBean.hermano.persona.tipoDoc}"
4 maxlength="9" />
```

Código 7.2: Componente Web de JSF

Como vemos, el `inputText` de JSF tiene como propiedades el atributo **required**, que indica la necesidad obligatoria de llevarlo cumplimentado en el formulario. Además, existe una segunda validación de longitud como es **maxlength** en la cual se indica en este caso que su longitud no puede superar los 9 caracteres.

En el caso de una validación mediante servidor, se realiza mediante la actualización del componente **growl**.

```
1 <p:growl id="growl" showDetail="true" life="3000">
2 <p:autoUpdate />
3 </p:growl>
```

Código 7.3: Growl de mensajes

Dicho componente puede ser autoactualizable por lo que funciona como un listener dentro de nuestro código. La funcionalidad que alimenta el contenido del growl se ubica en el bean y no es otra que la validación completa del formulario en su historia de usuario.

```

1  public void eliminarHermano() {
2      if (hermanos.findHermano(hermano)) {
3          Recibo reciboHermano = recibos.findByTipoReciboAndAñoAndHermano(
4              TipoRecibo.HERMANO, Util.getAñoActual(),
5              hermano);
6          Recibo reciboSalida = recibos.findByTipoReciboAndAñoAndHermano(TipoRecibo.SALIDA
7              , Util.getAñoActual(),
8              hermano);
9          if (reciboSalida != null) {
10             ControlesSalidas controlesSalidas = new ControlesSalidas();
11             ControlSalida control = controlesSalidas.findByHermanoAndTipoAndAño(hermano,
12                 TipoRecibo.SALIDA,
13                 Util.getAñoActual());
14             if (reciboSalida.getPagado().equals(Entidades.SI)) {
15                 if (control != null) {
16                     controlesSalidas.deleteControl(control);
17                 }
18                 recibos.deleteRecibo(reciboSalida);
19                 if (reciboHermano != null) {
20                     recibos.deleteRecibo(reciboHermano);
21                 }
22                 if (hermanos.eliminarHermano(hermano)) {
23                     Usuarios usuarios = new Usuarios();
24                     usuarios.borrar(hermano);
25                     if (hermano.getPersona().getEmail() != null) {
26                         mailClient = new MailClient(hermano.getPersona());
27                         mailClient.enviarCorreoHermano(MailClient.BAJA_HERMANO,
28                             hermano.getPersona(), null);
29                     }
30                     init();
31                     this.setHermano(new Hermano());
32                     this.addMessageInfo("Éxito en la operación", "Se ha dado de baja al
33                         hermano");
34                 } else {
35                     this.addMessageError("Error en la operación", "Ha existido un error al
36                         borrar");
37                 }
38             } else {
39                 this.addMessageError("Error en la operación",
40                     "El hermano tiene un control de salida sin pagar. Debe abonarlo antes de
41                         borrarlo de la nómina de hermanos.");
42             }
43         } else {
44             if (reciboHermano != null) {
45                 recibos.deleteRecibo(reciboHermano);
46             }
47             if (hermanos.eliminarHermano(hermano)) {
48                 Usuarios usuarios = new Usuarios();
49                 usuarios.borrar(hermano);
50                 if (hermano.getPersona().getEmail() != null) {
51                     mailClient = new MailClient(hermano.getPersona());
52                     mailClient.enviarCorreoHermano(MailClient.BAJA_HERMANO, hermano.getPersona
53                         (), null);
54                 }
55                 init();
56                 this.setHermano(new Hermano());
57                 this.addMessageInfo("Éxito en la operación", "Se ha dado de baja al hermano"
58                     );
59             } else {
60                 this.addMessageError("Error en la operación", "Ha existido un error al
61                     borrar");
62             }
63         }
64     } else {
65         this.addMessageError("Error en la operación", "No existe el hermano asociado a
66             ese DNI");
67     }
68 }

```

Código 7.4: EliminarHermano

Para cada uno de los mensajes que se añadan en el `addMessageError`, por detrás, existe una arquitectura creada indicando título, subtítulo, y gravedad del error, que a su vez va asociada al color del mensaje. Estos mensajes se muestran en pantalla cada vez que el usuario haga una persistencia en base de datos.

7.4. Métodos de interés

7.4.1. Verificación de DNI y predicción de letra

La herramienta cuenta con verificación para los campos de DNI introducidos en la aplicación. Esta función será llamada en varios puntos de la herramienta para comprobar si el DNI introducido dentro de la herramienta es válido o necesita de su revisión por parte del usuario.

Esto lo realiza por medio de la función `'validarNIEDNI'`, la cual usa como base la predicción de la letra para un documento de DNI ofrecido por el Ministerio del Interior y en base a ello, calcula si el DNI pertenece al estado español, o por el contrario, es extranjero.

```
1 public static boolean validarNIEDNI(final String ndoc) {
2     final String letras = "TRWAGMYFPDXBNJZSQVHLCKE";
3     Boolean res = true;
4     String str = ndoc.toString().toUpperCase();
5     if (str.length() == 10) {
6         str = str.replaceFirst("^0", "");
7     }
8     if (str.matches("^([0-9]{8}[TRWAGMYFPDXBNJZSQVHLCKE]{1})$")) {
9         res = (str.charAt(8) == letras.charAt(Integer.parseInt(str.substring(0, 8)) % 23));
10    } else if (str.matches("^([XYZ]{1}[0-9]{7}[TRWAGMYFPDXBNJZSQVHLCKE]{1})$")) {
11        str = str.replaceFirst("^X", "0").replaceFirst("^Y", "1").replaceFirst("^Z", "2");
12        res = (str.charAt(8) == letras.charAt(Integer.parseInt(str.substring(0, 8)) % 23));
13    } else {
14        res = false;
15    }
16    return res;
17 }
```

Código 7.5: validarNIEDNI()

7.4.2. Encriptación y descriptación de la contraseña

Para nuestra aplicación hemos creído conveniente el uso de contraseñas cifradas ya que cada usuario, tendrá la suya propia. Para ello, se utiliza el cifrado MD5 mediante Java y que en base a nuestra arquitectura, un administrador jamás puede saber la contraseña de un usuario ya que en MySQL se almacena cifrada.

```
1 public static String encriptar(String texto) {
2
3     String secretKey = "claveNOREALparalaDocumentacion"; // llave para encriptar datos
4     String base64EncryptedString = "";
5
6     try {
7
8         MessageDigest md = MessageDigest.getInstance("MD5");
9         byte[] digestOfPassword = md.digest(secretKey.getBytes("utf-8"));
10        byte[] keyBytes = Arrays.copyOf(digestOfPassword, 24);
11
12        SecretKey key = new SecretKeySpec(keyBytes, "DESede");
13        Cipher cipher = Cipher.getInstance("DESede");
14        cipher.init(Cipher.ENCRYPT_MODE, key);
15
16        byte[] plainTextBytes = texto.getBytes("utf-8");
```

```
17     byte[] buf = cipher.doFinal(plainTextBytes);
18     byte[] base64Bytes = Base64.encodeBase64(buf);
19     base64EncryptedString = new String(base64Bytes);
20
21     } catch (Exception ex) {
22     }
23     return base64EncryptedString;
24 }
```

Código 7.6: encriptar()

```

1 public static String desencriptar(String textoEncriptado) throws Exception {
2
3     String secretKey = "claveNOREALparalaDocumentacion"; // llave para encriptar datos
4     String base64EncryptedString = "";
5
6     try {
7         byte[] message = Base64.decodeBase64(textoEncriptado.getBytes("utf-8"));
8         MessageDigest md = MessageDigest.getInstance("MD5");
9         byte[] digestOfPassword = md.digest(secretKey.getBytes("utf-8"));
10        byte[] keyBytes = Arrays.copyOf(digestOfPassword, 24);
11        SecretKey key = new SecretKeySpec(keyBytes, "DESede");
12
13        Cipher decipher = Cipher.getInstance("DESede");
14        decipher.init(Cipher.DECRYPT_MODE, key);
15
16        byte[] plainText = decipher.doFinal(message);
17
18        base64EncryptedString = new String(plainText, "UTF-8");
19
20    } catch (Exception ex) {
21    }
22    return base64EncryptedString;
23 }

```

Código 7.7: desencriptar()

7.4.3. Envío de Email

Para una serie de acciones, la aplicación tiene la posibilidad de enviar un email al usuario con el que se tramita la acción. Tenemos diferentes acciones contempladas en el envío de un email:

- Alta de Hermano
- Baja de Hermano
- Restablecimiento de contraseña
- Edición de Hermano

Para la creación de una librería que haga uso del envío de email a los usuarios hemos decidido hacerlo en dos clases Java. Una que hará de Builder y otra que hará de Client, siendo la Builder la clase factoría que permita la creación de N clientes con diferentes propiedades.

```

1 package cofradia.presentation.helpers.correo;
2
3 public class MailBuilder {
4     private String host;
5     private String user;
6     private String clave;
7     private String port;
8
9     public MailBuilder() {
10        host = "smtp.gmail.com";
11        user = "cofradiatrabajofingrado@gmail.com";
12        clave = "CONTRASEÑAQUEHEMOSUSADO";
13        port = "587";
14    }
15
16    public String getHost() {
17        return host;
18    }
19
20    public void setHost(String host) {
21        this.host = host;
22    }
23
24    public String getUser() {
25        return user;
26    }

```

```
27
28 public void setUser(String user) {
29     this.user = user;
30 }
31
32 public String getClave() {
33     return clave;
34 }
35
36 public void setClave(String clave) {
37     this.clave = clave;
38 }
39
40 public String getPort() {
41     return port;
42 }
43
44 public void setPort(String port) {
45     this.port = port;
46 }
47 }
```

Código 7.8: Clase MailBuilder

Por otro lado, tenemos aquí la declaración de la clase MailClient.

```

1  package cofradia.presentation.helpers.correo;
2
3  import java.util.Properties;
4
5  import javax.activation.DataHandler;
6  import javax.activation.FileDataSource;
7  import javax.mail.BodyPart;
8  import javax.mail.Message;
9  import javax.mail.MessagingException;
10 import javax.mail.Session;
11 import javax.mail.Transport;
12 import javax.mail.internet.InternetAddress;
13 import javax.mail.internet.MimeBodyPart;
14 import javax.mail.internet.MimeMessage;
15 import javax.mail.internet.MimeMultipart;
16
17 import cofradia.persistencia.ControlSalida;
18 import cofradia.persistencia.Persona;
19
20 public class MailClient extends MailBuilder {
21
22     public static final String ALTA_HERMANO = "ALTA_HERMANO";
23     public static final String BAJA_HERMANO = "BAJA_HERMANO";
24     public static final String EDITAR_SALIDA = "EDITAR_SALIDA";
25     public static final String ALTA_SALIDA = "ALTA_SALIDA";
26     public static final String BAJA_SALIDA = "BAJA_SALIDA";
27     public static final String PASS = "PASS";
28     public static final String EDITAR = "EDITAR";
29     private Properties props = System.getProperties();
30     private Session session;
31     private MimeMessage message;
32     private BodyPart texto;
33     private BodyPart adjunto;
34
35     public MailClient(Persona persona) {
36         super();
37         session = Session.getDefaultInstance(props);
38         message = new MimeMessage(session);
39
40         props.put("mail.smtp.host", this.getHost()); // El servidor SMTP de Google
41         props.put("mail.smtp.user", persona.getEmail());
42         props.put("mail.smtp.clave", this.getClass()); // La clave de la cuenta
43         props.put("mail.smtp.auth", "true"); // Usar autenticación mediante usuario y
           clave
44         props.put("mail.smtp.starttls.enable", "true"); // Para conectar de manera
           segura al servidor SMTP
45         props.put("mail.smtp.port", this.getPort()); // El puerto SMTP seguro de Google
46     }
47
48     public MailClient(ControlSalida control) {
49         super();
50         session = Session.getDefaultInstance(props);
51         message = new MimeMessage(session);
52         texto = new MimeBodyPart();
53         adjunto = new MimeBodyPart();
54
55         props.put("mail.smtp.host", this.getHost()); // El servidor SMTP de Google
56         props.put("mail.smtp.user", control.getHermano().getPersona().getEmail());
57         props.put("mail.smtp.clave", this.getClass()); // La clave de la cuenta
58         props.put("mail.smtp.auth", "true"); // Usar autenticación mediante usuario y
           clave
59         props.put("mail.smtp.starttls.enable", "true"); // Para conectar de manera
           segura al servidor SMTP
60         props.put("mail.smtp.port", this.getPort()); // El puerto SMTP seguro de Google
61     }
62
63     public void enviarCorreoHermano(String asunto, Persona persona, String pass) {
64         String mensaje = new String();
65         try {

```

```

66     message.setFrom(new InternetAddress(this.getUser()));
67     message.addRecipient(Message.RecipientType.TO, new InternetAddress(
        persona.getEmail()));
68     switch (asunto) {
69         case ALTA_HERMANO:
70             asunto = "Alta de Hermano";
71             mensaje = crearMensajeAltaHermano(asunto, mensaje, persona, pass);
72             break;
73         case BAJA_HERMANO:
74             asunto = "Baja de Hermano";
75             mensaje = crearMensajeBajaHermano(asunto, mensaje, persona);
76             break;
77         case PASS:
78             asunto = "Se ha restablecido su contraseña";
79             mensaje = crearMensajeContraseña(asunto, mensaje, persona, pass);
80             break;
81         case EDITAR:
82             message.addRecipient(Message.RecipientType.TO, new InternetAddress(
                this.getUser()));
83             asunto = "Se ha editado el hermano";
84             mensaje = crearMensajeEditarHermano(asunto, mensaje, persona);
85             break;
86     }
87     message.setSubject(asunto);
88     message.setText(mensaje);
89
90     Transport transport = session.getTransport("smtp");
91     transport.connect(this.getHost(), this.getUser(), this.getClave());
92     transport.sendMessage(message, message.getAllRecipients());
93     transport.close();
94 } catch (MessagingException me) {
95     me.printStackTrace(); // Si se produce un error
96 }
97 }
98
99 public String crearMensajeEditarHermano(String asunto, String mensaje, Persona
    persona) {
100     mensaje = "VENERABLE COFRADÍA DE PENITENCIA DEL STMO. CRISTO DEL PERDÓN Y MARIA
        STMA. DEL ROSARIO EN SUS MISTERIOS DOLOROSOS\n\n";
101
102     mensaje = mensaje.concat("El hermano con DNI ").concat(persona.getDni().concat(
103         " ha editado el método de pago de su cuota. Revíselo para añadirlo a la orden de
        pago del banco.\n\n"));
104
105     return mensaje;
106 }
107
108 public String crearMensajeContraseña(String asunto, String mensaje, Persona
    persona, String pass) {
109     mensaje = "VENERABLE COFRADÍA DE PENITENCIA DEL STMO. CRISTO DEL PERDÓN Y MARIA
        STMA. DEL ROSARIO EN SUS MISTERIOS DOLOROSOS\n\n";
110
111     mensaje = mensaje.concat("Hemos recibido una petición para restablecer su
        contraseña\n\n");
112
113     mensaje = mensaje.concat("Sus credenciales para iniciar sesión en el portal como
        hermano son: \n\n Usuario: "
114         .concat(persona.getDni().concat("\n Contraseña: ").concat(pass).concat("\n\n"));
115
116     mensaje = mensaje.concat(
117         "Nos alegra que esté entre nuestra mas ilustre y venerada familia. Gracias.\n\n"
        + "Tu Hermano Mayor");
118
119     return mensaje;
120 }
121
122 public void enviarCorreoSalida(String asunto, ControlSalida control) {
123     String mensaje = new String();
124     MimeMultipart multiParte = new MimeMultipart();

```



```

125     try {
126         message.setFrom(new InternetAddress(this.getUser()));
127         message.addRecipient(Message.RecipientType.TO,
128             new InternetAddress(control.getHermano().getPersona().getEmail()));
129         switch (asunto) {
130             case ALTA_SALIDA:
131                 asunto = "Alta de Control de Salida";
132                 mensaje = crearMensajeAltaSalida(asunto, mensaje, control);
133
134                 message.setSubject(asunto);
135                 texto.setText(mensaje);
136                 adjunto.setDataHandler(
137                     new DataHandler(new FileDataSource("/Users/yisekai/Downloads/PDF JAVA/
JustificanteSalida_"
138                         .concat(control.getHermano().getPersona().getDni()).concat(".pdf"))));
139                 adjunto.setFileName(
140                     "JustificanteSalida_" .concat(control.getHermano().getPersona().getDni()).
concat(".pdf"));
141
142                 multiParte.addBodyPart(texto);
143                 multiParte.addBodyPart(adjunto);
144
145                 // Se mete el texto y la foto adjunta.
146                 message.setContent(multiParte);
147
148                 break;
149             case EDITAR_SALIDA:
150                 asunto = "Alta de Control de Salida";
151                 mensaje = crearMensajeEditarSalida(asunto, mensaje, control);
152
153                 message.setSubject(asunto);
154                 texto.setText(mensaje);
155                 adjunto.setDataHandler(
156                     new DataHandler(new FileDataSource("/Users/yisekai/Downloads/PDF JAVA/
JustificanteSalida_"
157                         .concat(control.getHermano().getPersona().getDni()).concat(".pdf"))));
158                 adjunto.setFileName(
159                     "JustificanteSalida_" .concat(control.getHermano().getPersona().getDni()).
concat(".pdf"));
160
161                 multiParte.addBodyPart(texto);
162                 multiParte.addBodyPart(adjunto);
163
164                 // Se mete el texto y la foto adjunta.
165                 message.setContent(multiParte);
166
167                 break;
168             case BAJA_SALIDA:
169                 asunto = "Baja de Control de Salida";
170                 mensaje = crearMensajeBajaSalida(asunto, mensaje, control);
171
172                 message.setSubject(asunto);
173                 mensaje.setText(mensaje);
174                 break;
175         }
176
177         Transport transport = session.getTransport("smtp");
178         transport.connect(this.getHost(), this.getUser(), this.getClave());
179         transport.sendMessage(message, message.getAllRecipients());
180         transport.close();
181     } catch (MessagingException me) {
182         me.printStackTrace(); // Si se produce un error
183     }
184
185 }
186
187 private String crearMensajeAltaHermano(String asunto, String mensaje, Persona
persona, String pass) {
188     asunto = "Alta de Hermano";
189     mensaje = "VENERABLE COFRADÍA DE PENITENCIA DEL STMO. CRISTO DEL PERDÓN Y MARIA

```

```

    STMA. DEL ROSARIO EN SUS MISTERIOS DOLOROSOS\n\n";
190 mensaje = mensaje.concat("Se ha dado de alta como hermano a N.H.D ");
191 .concat(persona.getNombre().concat(" ").concat(persona.getApellido1()));
192 if (persona.getApellido2() != null) {
193     mensaje = mensaje.concat(" ").concat(persona.getApellido2());
194 }
195 mensaje = mensaje.concat(" ").concat(" con DNI ").concat(persona.getDni())
196 .concat("\n\n" + "Sus credenciales para iniciar sesión en el portal como hermano
    son: \n\n Usuario: "
197 .concat(persona.getDni()).concat("\n Contraseña: ").concat(pass).concat("\n\n"));
    ;
198
199 mensaje = mensaje.concat(
200     "Nos alegra que esté entre nuestra mas ilustre y venerada familia. Gracias.\n\n"
    + "Tu Hermano Mayor");
201 return mensaje;
202 }
203
204 private String crearMensajeBajaHermano(String asunto, String mensaje, Persona
    persona) {
205     asunto = "Baja de Hermano";
206     mensaje = mensaje.concat(
207         "VENERABLE COFRADÍA DE PENITENCIA DEL STMO. CRISTO DEL PERDÓN Y MARIA STMA. DEL
    ROSARIO EN SUS MISTERIOS DOLOROSOS\n\n"
208     + "Se ha dado de baja como hermano a N.H.D "
209     .concat(persona.getNombre().concat(" ").concat(persona.getApellido1()));
210     if (persona.getApellido2() != null) {
211         mensaje = mensaje.concat(" ").concat(persona.getApellido2());
212     }
213     mensaje = mensaje.concat(" ").concat(" con DNI ").concat(persona.getDni()).
    concat("\n\n"
214     + "Sentimos su baja como hermano. Le deseo toda la suerte y amor del mundo. Aquí
    tendrás por siempre a tu Hermandad. Gracias por todo.\n\n"
215     + "Tu Hermano Mayor");
216     return mensaje;
217 }
218
219 private String crearMensajeAltaSalida(String asunto, String mensaje, ControlSalida
    control) {
220     asunto = control.getProcesion().getDescripcion();
221     mensaje = mensaje.concat(
222         "VENERABLE COFRADÍA DE PENITENCIA DEL STMO. CRISTO DEL PERDÓN Y MARIA STMA. DEL
    ROSARIO EN SUS MISTERIOS DOLOROSOS\n\n"
223     + "Se ha dado de alta el control de salida de N.H.D ".concat(control.getHermano
    ().getPersona()
224     .getNombre().concat(" ").concat(control.getHermano().getPersona().getApellido1()
    ));
225     if (control.getHermano().getPersona().getApellido2() != null) {
226         mensaje = mensaje.concat(" ").concat(control.getHermano().getPersona().
    getApellido2());
227     }
228     mensaje = mensaje.concat(" ").concat(" con DNI ").concat(control.getHermano().
    getPersona().getDni()).concat(
229     "\n\n" + "Espero que disfrutes de tus titulares durante todo el
    recorrido.Contigo somos Hermandad. Gracias.\n\n"
230     + "Tu Hermano Mayor");
231     return mensaje;
232 }
233 }
234
235 private String crearMensajeEditarSalida(String asunto, String mensaje,
    ControlSalida control) {
236     asunto = control.getProcesion().getDescripcion();
237     mensaje = mensaje.concat(
238         "VENERABLE COFRADÍA DE PENITENCIA DEL STMO. CRISTO DEL PERDÓN Y MARIA STMA. DEL
    ROSARIO EN SUS MISTERIOS DOLOROSOS\n\n"
239     + "Se ha editado el control de salida de N.H.D ".concat(control.getHermano().
    getPersona()
240     .getNombre().concat(" ").concat(control.getHermano().getPersona().getApellido1()
    ));

```

```

241     if (control.getHermano().getPersona().getApellido2() != null) {
242         mensaje = mensaje.concat(" ").concat(control.getHermano().getPersona().
            getApellido2());
243     }
244     mensaje = mensaje.concat(" ").concat(" con DNI ").concat(control.getHermano().
        getPersona().getDni())
245     .concat("\n\n");
246     if (control.getSeccion() != null) {
247         mensaje = mensaje.concat("Se le ha asignado la sección ").concat(
            control.getSeccion().concat(" del paso "));
248
249         if (control.getHermano().getTunica().getPaso().getDescripcion().equals("CRISTO
            ")) {
250             mensaje = mensaje.concat("de nuestro Santísimo Cristo del Perdón\n\n");
251         } else {
252             mensaje = mensaje.concat("de nuestra María Santísima del Rosario en sus
                Misterios Dolorosos\n\n");
253         }
254     }
255     mensaje = mensaje.concat(
256         "Espero que disfrutes de tus titulares durante todo el recorrido.Contigo somos
            Hermandad. Gracias.\n\n"
257         + "Tu Hermano Mayor");
258     return mensaje;
259 }
260
261 private String crearMensajeBajaSalida(String asunto, String mensaje, ControlSalida
    control) {
262     asunto = control.getProcesion().getDescripcion();
263     mensaje = "VENERABLE COFRADÍA DE PENITENCIA DEL STMO. CRISTO DEL PERDÓN Y MARIA
        STMA. DEL ROSARIO EN SUS MISTERIOS DOLOROSOS\n\n Se ha dado de baja el
        control de salida de N.H.D "
264         .concat(control.getHermano().getPersona().getNombre().concat(" ")
            .concat(control.getHermano().getPersona().getApellido1()));
265     if (control.getHermano().getPersona().getApellido2() != null) {
266         mensaje = mensaje.concat(" ").concat(control.getHermano().getPersona().
            getApellido2());
267     }
268     mensaje = mensaje.concat(" ").concat(" con DNI ").concat(control.getHermano().
        getPersona().getDni()).concat(
269         "\n\n" + "Esperamos que para el año que viene contemos con tu asistencia. Los
            hermanos somos el mayor patrimonio de la Hermandad. Gracias por todo.\n\n"
270         + "Tu Hermano Mayor");
271     return mensaje;
272 }
273
274 public Properties getProps() {
275     return props;
276 }
277
278 public void setProps(Properties props) {
279     this.props = props;
280 }
281
282 public Session getSession() {
283     return session;
284 }
285
286 public void setSession(Session session) {
287     this.session = session;
288 }
289
290 public MimeMessage getMessage() {
291     return message;
292 }
293
294 public void setMessage(MimeMessage message) {
295     this.message = message;
296 }

```

```
299     }  
300 }
```

Código 7.9: Clase MailClient

7.4.3.1. Documentación personalizada

A la hora de crear un Control de Salida, de acuerdo con el cliente, se ha establecido la creación parametrizada de un documento PDF que será enviado mediante mail al hermano. Dicha modelación del documento se ha creado en Java usando la librería externa **iTextPdf**.

```

1  package cofradia.presentation.helpers.pdf;
2
3  import java.io.FileOutputStream;
4  import java.util.Date;
5  import java.util.Locale;
6
7  import org.primefaces.model.StreamedContent;
8
9  import com.itextpdf.text.BaseColor;
10 import com.itextpdf.text.Chunk;
11 import com.itextpdf.text.Document;
12 import com.itextpdf.text.Font;
13 import com.itextpdf.text.Font.FontFamily;
14 import com.itextpdf.text.Image;
15 import com.itextpdf.text.PageSize;
16 import com.itextpdf.text.Paragraph;
17 import com.itextpdf.text.pdf.PdfWriter;
18
19 import cofradia.base.Entidades;
20 import cofradia.base.Util;
21 import cofradia.persistencia.ControlSalida;
22 import cofradia.persistencia.Usuario;
23
24 public class BuilderPdf {
25
26     public static final String textol = "VENERABLE COFRADÍA DE PENITENCIA DEL STMO.
27         CRISTO DEL PERDÓN Y MARIA STMA. DEL ROSARIO EN SUS MISTERIOS DOLOROSOS\n\n";
28     private PdfHelpers pdfHelpers;
29     private StreamedContent content;
30     private ControlSalida control;
31
32     public BuilderPdf(ControlSalida control, String rol) {
33         this.control = control;
34         pdfControlSalida(control, rol);
35     }
36
37     public void pdfControlSalida(ControlSalida control, String rol) {
38         try {
39             Document document = new Document(PageSize.A4);
40             FileOutputStream salida = new FileOutputStream("/Users/yisekai/Downloads/PDF
41                 JAVA/JustificanteSalida_"
42             .concat(control.getHermano().getPersona().getDni()).concat(".pdf"));
43             PdfWriter writer = PdfWriter.getInstance(document, salida);
44             writer.setInitialLeading(0);
45             // writer.setPageEvent(pdfHelpers);
46             document.open();
47             // Imagen 1
48             Image image = Image.getInstance(this.getClass().getClassLoader().getResource("
49                 images/pdfTop.png"));
50             image.scaleToFit(600, 300);
51             image.setAlignment(Chunk.ALIGN_CENTER);
52             document.add(image);
53
54             Font f = new Font(FontFamily.HELVETICA, 20.0f, Font.BOLDITALIC,
55                 BaseColor.BLACK);
56             Paragraph p = new Paragraph(textol, f);
57
58             p.setAlignment(Paragraph.ALIGN_CENTER);
59
60             // Texto
61             document.add(p);
62
63             Font f1 = new Font(FontFamily.HELVETICA, 14.0f, Font.NORMAL, BaseColor.BLACK);
64             Paragraph p1;

```

```

61     if (control.getHermano().getPersona().getApellido2() != null) {
62         if (rol.equals(Usuario.ADMINISTRADOR)) {
63             p1 = new Paragraph("N.H.D ".concat(control.getHermano().getPersona().
64                 getNombre()).concat(" ")
65                 .concat(control.getHermano().getPersona().getApellido1()).concat(" ")
66                 .concat(control.getHermano().getPersona().getApellido2()).concat(" con
67                     DNI: ")
68                 .concat(control.getHermano().getPersona().getDni())
69                 .concat(" acompañará a nuestros amados titulares en la salida procesional
70                     del ").concat(Util
71                     .dateToString(control.getProcesion().getFecha(), Entidades.DEFAULT_PATTERN
72                     ,
73                     Locale.getDefault()))
74                 .concat(" portando ").concat(control.getOcupacion().getDescripcion())
75                 .concat(".\n\n"))),
76             f1);
77         } else {
78             p1 = new Paragraph("N.H.D ".concat(control.getHermano().getPersona().
79                 getNombre()).concat(" ")
80                 .concat(control.getHermano().getPersona().getApellido1()).concat(" ")
81                 .concat(control.getHermano().getPersona().getApellido2()).concat(" con
82                     DNI: ")
83                 .concat(control.getHermano().getPersona().getDni())
84                 .concat(" acompañará a nuestros amados titulares en la salida procesional
85                     del ")
86                 .concat(Util.dateToString(control.getProcesion().getFecha(),
87                     Entidades.DEFAULT_PATTERN, Locale.getDefault()))),
88             f1);
89         }
90     } else {
91         if (rol.equals(Usuario.ADMINISTRADOR)) {
92             p1 = new Paragraph(
93                 "N.H.D ".concat(control.getHermano().getPersona().getNombre()).concat(" ")
94                 .concat(control.getHermano().getPersona().getApellido1()).concat(" con
95                     DNI: ")
96                 .concat(control.getHermano().getPersona().getDni())
97                 .concat(" acompañará a nuestros amados titulares en la salida procesional
98                     del ")
99                 .concat(Util
100                     .dateToString(control.getProcesion().getFecha(),
101                     Entidades.DEFAULT_PATTERN, Locale.getDefault()))
102                 .concat(" portando ")
103                 .concat(control.getOcupacion().getDescripcion()).concat(".\n\n")),
104             f1);
105         } else {
106             p1 = new Paragraph("N.H.D ".concat(control.getHermano().getPersona().
107                 getNombre()).concat(" ")
108                 .concat(control.getHermano().getPersona().getApellido1()).concat(" con
109                     DNI: ")
110                 .concat(control.getHermano().getPersona().getDni())
111                 .concat(" acompañará a nuestros amados titulares en la salida procesional
112                     del ")
113                 .concat(Util.dateToString(control.getProcesion().getFecha(),
114                     Entidades.DEFAULT_PATTERN, Locale.getDefault()))),
115             f1);
116         }
117     }
118
119     p1.setAlignment(Paragraph.ALIGN_CENTER);
120
121     // Texto
122     document.add(p1);
123
124     Font f2 = new Font(FontFamily.HELVETICA, 14.0f, Font.NORMAL, BaseColor.BLACK);
125     Paragraph p2 = new Paragraph(
126         "En Cádiz a ".concat(Util.dateToString(new Date(), Entidades.DEFAULT_PATTERN,
127             Locale.getDefault()))
128         .concat(".\n\n"),
129         f2);

```

```

118
119     p2.setAlignment(Paragraph.ALIGN_CENTER);
120     // Texto
121     document.add(p2);
122
123     // texto
124     Font f3 = new Font(FontFamily.HELVETICA, 14.0f, Font.BOLD, BaseColor.BLACK);
125     Paragraph p3 = new Paragraph("EL MAYORDOMO", f3);
126
127     p3.setAlignment(Paragraph.ALIGN_CENTER);
128     // Texto
129     document.add(p3);
130     // Imagen 2
131     Image imageDown = Image.getInstance(this.getClass().getClassLoader().
132         getResource("images/pdfDown.png"));
133     imageDown.scaleToFit(600, 300);
134     imageDown.setAlignment(Chunk.ALIGN_CENTER);
135     imageDown.setAbsolutePosition(0f, 0f);
136     document.add(imageDown);
137     document.close();
138 } catch (Exception e) {
139     e.printStackTrace();
140 }
141
142
143 public String getNombreDocumento() {
144     if (control.getHermano().getPersona().getDni() != null) {
145         return "JustificanteSalida_".concat(control.getHermano().getPersona().getDni())
146             .concat(".pdf");
147     } else {
148         return "default.pdf";
149     }
150 }
151
152 public PdfHelpers getPdfHelpers() {
153     return pdfHelpers;
154 }
155
156 public void setPdfHelpers(PdfHelpers pdfHelpers) {
157     this.pdfHelpers = pdfHelpers;
158 }
159
160 public StreamedContent getContent() {
161     return content;
162 }
163
164 public void setContent(StreamedContent content) {
165     this.content = content;
166 }
167
168 public ControlSalida getControl() {
169     return control;
170 }
171
172 public void setControl(ControlSalida control) {
173     this.control = control;
174 }

```

Código 7.10: Clase BuilderPDF

7.4.4. Pago mediante PayPal

El pago mediante PayPal me ha resultado el módulo mas difícil de desarrollar dentro de la aplicación por total desconocimiento del mismo. La facilidad de poder contar con un entorno de pruebas donde apuntar con nuestro servicio REST facilitó el trabajo pero, no obstante, creo que este apartado marca un antes y después en el desarrollo del proyecto.

Para desarrollar dicho módulo planteamos la solución con una arquitectura vertical para hacer una factoría de objetos sobre el mismo de la siguiente forma:

1. PayPalBuilder Esta clase contiene las propiedades de PayPal, entre las que destacamos las siguientes:
 - Divisa
 - Metodo
 - Intent
 - Entorno
 - clientID
 - clientSecret
2. PayPalClient Esta clase debe heredar de PayPalBuilder y tendrá como objetos propios, necesarios para el uso de la API, las siguientes propiedades:
 - Amount
 - Payer
 - Payment
 - RedirectUrls
 - List<Transaction>


```
1 package cofradia.presentation.helpers.paypal;
2
3 public class PayPalBuilder{
4
5     private final String divisa = "EUR";
6     private final String metodo = "paypal";
7     private final String intent = "sale";
8     private final String entorno = "sandbox";
9
10    private final String clientId = "clientID_USADA";
11    private final String clientSecret = "clientSecret_USADA";
12
13    public PayPalBuilder() {
14
15    }
16
17    public String getDivisa() {
18        return divisa;
19    }
20
21    public String getMetodo() {
22        return metodo;
23    }
24
25    public String getIntent() {
26        return intent;
27    }
28
29    public String getClientId() {
30        return clientId;
31    }
32
33    public String getClientSecret() {
34        return clientSecret;
35    }
36
37    public String getEntorno() {
38        return entorno;
39    }
40
41
42 }
```

Código 7.11: Clase PayPalBuilder

```
1 package cofradia.presentation.helpers.paypal;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import javax.annotation.PostConstruct;
7
8 import com.paypal.api.payments.Amount;
9 import com.paypal.api.payments.Payer;
10 import com.paypal.api.payments.Payment;
11 import com.paypal.api.payments.RedirectUrls;
12 import com.paypal.api.payments.Transaction;
13 import com.paypal.base.rest.APIContext;
14 import com.paypal.base.rest.PayPalRESTException;
15
16 public class PayPalClient extends PayPalBuilder {
17
18     private Amount amount;
19     private Payer payer;
20     private Transaction transaction;
21     private List<Transaction> transactions;
22     private Payment payment;
23     private RedirectUrls redirectUrls;
24     private String cantidad;
25     private String urlCancel;
26     private String urlSuccess;
27
28     public PayPalClient(final String cantidad, String urlCancel, String urlSuccess) {
29         super();
30         this.cantidad = cantidad;
31         this.urlCancel = urlCancel;
32         this.urlSuccess = urlSuccess;
33
34         amount = new Amount();
35         amount.setCurrency(this.getDivisa());
36         amount.setTotal(cantidad);
37
38         transaction = new Transaction();
39         transaction.setAmount(amount);
40         transactions = new ArrayList<Transaction>();
41         transactions.add(transaction);
42
43         payer = new Payer();
44         payer.setPaymentMethod(this.getMetodo());
45
46         payment = new Payment();
47         payment.setIntent(this.getIntent());
48         payment.setPayer(payer);
49         payment.setTransactions(transactions);
50
51         redirectUrls = new RedirectUrls();
52         redirectUrls.setCancelUrl(this.getUrlCancel());
53         redirectUrls.setReturnUrl(this.getUrlSuccess());
54
55         payment.setRedirectUrls(redirectUrls);
56     }
57
58     @PostConstruct
59     public void init() {
60     }
61
62     public boolean iniciarPago() {
63         boolean var = false;
64         try {
65             APIContext apiContext = new APIContext(this.getClientId(), this.getClientSecret(),
66                 this.getEntorno());
67             Payment createdPayment = payment.create(apiContext);
68             var = true;
69             System.out.println(createdPayment.toString());
70         } catch (PayPalRESTException e) {
```

```
70 var = false;
71 e.printStackTrace();
72 } catch (Exception ex) {
73 var = false;
74 ex.printStackTrace();
75 }
76 return var;
77 }
78
79 public Amount getAmount() {
80 return amount;
81 }
82
83 public void setAmount(Amount amount) {
84 this.amount = amount;
85 }
86
87 public Payer getPayer() {
88 return payer;
89 }
90
91 public void setPayer(Payer payer) {
92 this.payer = payer;
93 }
94
95 public Transaction getTransaction() {
96 return transaction;
97 }
98
99 public void setTransaction(Transaction transaction) {
100 this.transaction = transaction;
101 }
102
103 public List<Transaction> getTransactions() {
104 return transactions;
105 }
106
107 public void setTransactions(List<Transaction> transactions) {
108 this.transactions = transactions;
109 }
110
111 public Payment getPayment() {
112 return payment;
113 }
114
115 public void setPayment(Payment payment) {
116 this.payment = payment;
117 }
118
119 public RedirectUrls getRedirectUrls() {
120 return redirectUrls;
121 }
122
123 public void setRedirectUrls(RedirectUrls redirectUrls) {
124 this.redirectUrls = redirectUrls;
125 }
126
127 public String getCantidad() {
128 return cantidad;
129 }
130
131 public void setCantidad(String cantidad) {
132 this.cantidad = cantidad;
133 }
134
135 public String getUrlCancel() {
136 return urlCancel;
137 }
138
139 public void setUrlCancel(String urlCancel) {
```

```
140 this.urlCancel = urlCancel;
141 }
142
143 public String getUrlSuccess() {
144     return urlSuccess;
145 }
146
147 public void setUrlSuccess(String urlSuccess) {
148     this.urlSuccess = urlSuccess;
149 }
150
151 }
```

Código 7.12: Clase PayPalClient

Dando como resultado positivo el siguiente XML de la API de PayPal y que es interesante tener en cuenta:

```
1  {
2      "id": "PAYID-L3FFVIY4NE969028J136940U",
3      "intent": "sale",
4      "payer": {
5          "payment_method": "paypal"
6      },
7      "transactions": [
8          {
9              "related_resources": [],
10             "amount": {
11                 "currency": "EUR",
12                 "total": "32.00"
13             }
14         }
15     ],
16     "state": "created",
17     "create_time": "2020-05-24T11:29:38Z",
18     "links": [
19         {
20             "href": "https://api.sandbox.paypal.com/v1/payments/payment/PAYID-
21                 L3FFVIY4NE969028J136940U",
22             "rel": "self",
23             "method": "GET"
24         },
25         {
26             "href": "https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_u003d_express-
27                 checkout\u0026token\u003dEC-5N444608KR503192U",
28             "rel": "approval_url",
29             "method": "REDIRECT"
30         },
31         {
32             "href": "https://api.sandbox.paypal.com/v1/payments/payment/PAYID-
33                 L3FFVIY4NE969028J136940U/execute",
34             "rel": "execute",
35             "method": "POST"
36         }
37     ]
38 }
```

Código 7.13: XML API PayPal

7.5. Accesos a la Base de Datos

Como ya hemos explicado en otras secciones del documento, el desarrollo lo hemos realizado con el IDE Eclipse, y para los accesos a Base de Datos con MySQL hemos usado un Driver SQL que permite la conexión de este IDE con la BD.

Como anécdota, mi aplicación sufría retardos y timeouts en los accesos y consultas a BD y resultó ser por la variable db de la clase. Al no ser estática, cada entidad que la invocaba hacía la suya propia creando para ella su propio hilo. Ésto se solucionó comprobando el valor de la variable en su init y haciendo a la variable estática para el acceso común del recurso a las otras entidades.

Para crear la arquitectura, hemos decidido crear una clase de la cual heredan las demás. En este caso, DataBaseBuilder.java.

```
1
2 package cofradia.base;
3
4 import java.sql.Connection;
5 import java.sql.DriverManager;
6
7 import javax.faces.bean.RequestScoped;
8 import javax.inject.Named;
9
10 /**
11  * Clase que configura la conexión a la base de datos
12  * @author jmgarcial
13  */
14 @Named
15 @RequestScoped
16 public class DataBaseBuilder {
17
18     /**
19      * Variable estática que aporta información
20      */
21     public static final String INSERT = "INSERT INTO COFRADIA.";
22
23     /**
24      * Variable estática que aporta información
25      */
26     public static final String UPDATE = "UPDATE TABLE COFRADIA.";
27
28     /**
29      * Variable estática que aporta información
30      */
31     public static final String SELECT = "SELECT * FROM COFRADIA.";
32
33     /**
34      * Variable estática que aporta información
35      */
36     public static final String FROM = " FROM COFRADIA.";
37
38     /**
39      * Variable estática que aporta información
40      */
41     public static final String WHERE = " WHERE ";
42
43     /**
44      * Variable estática que aporta información
45      */
46     public static final String DELETE = "DELETE FROM COFRADIA.";
47
48     /**
49      * Variable estática que aporta información
50      */
51     public static final String EQUALS = " = ";
52
53     /**
54      * Variable estática que aporta información
55      */
```

```
56     public static final String IN = " IN ";
57
58     /**
59     * Variable estática que aporta información
60     */
61     public static final String SELECT_FK = "(SELECT ";
62
63     /**
64     * Variable estática que aporta información
65     */
66     public final String LOCALHOST = "localhost";
67
68     /**
69     * Variable estática que aporta información
70     */
71     public final String PORT = "8889";
72
73     /**
74     * Variable estática que aporta información
75     */
76     public final String DATABASE = "COFRADIA";
77
78     /**
79     * Variable estática que aporta información
80     */
81     public final String USER = "root";
82
83     /**
84     * Variable estática que aporta información
85     */
86     public final String PASS = "root";
87
88     public static Connection db;
89
90     private String host;
91
92     private String port;
93
94     private String database;
95
96     private String user;
97
98     private String password;
99
100    /**
101    * Constructor
102    */
103    public DataBaseBuilder() {
104        this.host = LOCALHOST;
105        this.port = PORT;
106        this.database = DATABASE;
107        this.user = USER;
108        this.password = PASS;
109        this.init();
110    }
111
112    public void init() {
113        if (db == null) {
114            db = this.connectDatabase(host, port, database, user, password);
115        }
116    }
117
118    /**
119    * Metodo para realizar una conexión a la base de datos
120    * @param host
121    * @param port
122    * @param database
123    * @param user
124    * @param password
125    * @return variable con la conexión realizada
```

```
126  */
127  public Connection connectDatabase(String host, String port, String database,
    String user, String password) {
128      Connection connection = null;
129      boolean valid = false;
130      String url = null;
131      try {
132          // We register the MySQL driver
133          // Registramos el driver de MySQL
134          try {
135              Class.forName("com.mysql.jdbc.Driver");
136          } catch (ClassNotFoundException ex) {
137              System.out.println("Error al registrar el driver de MySQL: " + ex);
138          }
139          url = "jdbc:mysql://" + host + ":" + port + "/" + database + "?useSSL=false";
140          // Database connect
141          // Conectamos con la base de datos
142          connection = DriverManager.getConnection(url, user, password);
143          valid = connection.isValid(50000);
144      } catch (java.sql.SQLException sqle) {
145          System.out.println("Error al conectar con la base de datos de MySQL (" + url +
            "): " + sqle);
146      }
147      return connection;
148  }
149
150  /**
151   * Getter/Setter
152   * @return
153   */
154  public Connection getDb() {
155      return db;
156  }
157
158  /**
159   * Getter/Setter
160   * @return
161   */
162  public String getHost() {
163      return host;
164  }
165
166  /**
167   * Getter/Setter
168   * @return
169   */
170  public void setHost(String host) {
171      this.host = host;
172  }
173
174  /**
175   * Getter/Setter
176   * @return
177   */
178  public String getPort() {
179      return port;
180  }
181
182  /**
183   * Getter/Setter
184   * @return
185   */
186  public void setPort(String port) {
187      this.port = port;
188  }
189
190  /**
191   * Getter/Setter
192   * @return
193   */
```

```
194     public String getDatabase() {
195         return database;
196     }
197
198     /**
199     * Getter/Setter
200     * @return
201     */
202     public void setDatabase(String database) {
203         this.database = database;
204     }
205
206     /**
207     * Getter/Setter
208     * @return
209     */
210     public String getUser() {
211         return user;
212     }
213
214     /**
215     * Getter/Setter
216     * @return
217     */
218     public void setUser(String user) {
219         this.user = user;
220     }
221
222     /**
223     * Getter/Setter
224     * @return
225     */
226     public String getPassword() {
227         return password;
228     }
229
230     /**
231     * Getter/Setter
232     * @return
233     */
234     public void setPassword(String password) {
235         this.password = password;
236     }
237
238     /**
239     * Getter/Setter
240     * @return
241     */
242     public void setDb(Connection db) {
243         this.db = db;
244     }
245 }
```

Código 7.14: Clase DataBaseBuilder

Capítulo 8

Pruebas y validación

En este capítulo se expondrán las diversas pruebas que se han realizado sobre la herramienta y la forma en que estas han sido enfocadas para obtener los resultados esperados y cumplir con los objetivos fijados.

8.1. Pruebas unitarias

Para la realización de las pruebas unitarias del aplicativo se ha utilizado JUnit y la librería Mocks, las cuales han facilitado la creación de diversas pruebas aisladas sobre cada funcionalidad de la aplicación. Se han realizado las siguientes pruebas sobre los módulos:

- Unitarias
 - Control de Salida
- Automáticas
 - Alta de Persona en el sistema
 - Baja de Persona en el sistema
 - Alta de Hermano en el sistema
 - Baja de Hermano en el sistema
 - Alta de Control de Salida en el sistema
 - Baja de Control de Salida en el sistema
 - Alta de Túnica en el sistema
 - Baja de Túnica en el sistema

Todas estas pruebas, sobre acciones individuales que realiza la herramienta, constatan un total de 9 pruebas que comprueban las funcionalidades base de los objetivos fijados en el capítulo de análisis sobre las funcionalidades que debía poseer la herramienta final.

Estos tests han sido implementados a medida que la herramienta incrementaba sus funcionalidades, por lo que han sido probados en diversas ocasiones para comprobar si dichas pruebas verifican un buen funcionamiento de la herramienta y de las diversas acciones de la herramienta que estas pruebas validan.

A continuación, se muestran algunos ejemplos de tests unitarios de la herramienta:

```
1 package cofradia.auto;  
2  
3 import org.junit.Test;  
4 import org.openqa.selenium.By;
```

```
5  import org.openqa.selenium.support.ui.Select;
6
7  public class AutoHermanos extends Login{
8
9      @Test
10     public void altaHermano() throws InterruptedException {
11         this.getDriver().get(ADD_HERMANOS);
12         this.getDriver().findElement(By.id("form:dniBusqueda")).sendKeys("Z6546378P");
13         this.getDriver().findElement(By.id("form:btnBuscar")).click();
14         Thread.sleep(2000);
15         Select select = new Select(this.getDriver().findElement(By.id("form:recibo")));
16         select.selectByValue("2");
17         this.getDriver().findElement(By.id("form:btnAltaHermano")).click();
18         Thread.sleep(2000);
19         this.getDriver().get(LIST_HERMANOS);
20         this.getDriver().close();
21     }
22
23     @Test
24     public void bajaHermano() throws InterruptedException {
25         this.getDriver().get(DELETE_HERMANOS);
26         this.getDriver().findElement(By.id("form:dniBusqueda")).sendKeys("Z6546378P");
27         Thread.sleep(1000);
28         this.getDriver().findElement(By.id("form:btnBuscar")).click();
29         Thread.sleep(1000);
30         this.getDriver().get(LIST_HERMANOS);
31         this.getDriver().close();
32     }
33 }
```

Código 8.1: Clase AutoHermanos.java

8.2. Pruebas de integración

Durante el desarrollo de la herramienta, se han ido implementando los diversos módulos por separado, realizándose diversas pruebas de integración para las acciones que estos poseen, donde debido a la forma de programación de Java, se realizó en una dificultad ascendente.

Esto quiere decir que se comenzó integrando las diversas vistas para los tipos de usuarios, sin funcionalidades en las mismas, y más adelante, incluyendo una a una las acciones requeridas para cada módulo, coincidiendo así, con el desarrollo de pruebas unitarias para dichas acciones.

8.3. Pruebas de aceptación

La herramienta ha sido expuesta frente a sus usuarios finales a lo largo de su desarrollo en diversas etapas, donde estos, expresaban su opinión sobre la misma y las posibles carencias o necesidades que precisaban en el momento. Esto propició la implementación de acciones no reflejadas en los objetivos iniciales, dando como lugar, fijar los objetivos finales que se encuentran en el apartado correspondiente del capítulo de calendario.

Por ello, la herramienta se encuentra en la versión 1.0, la cual, es la versión final de la aplicación que cumple con los objetivos finales de alcance que me propuse para la entrega del TFG con la Universidad de Cádiz y las posteriores versiones con mejoras tendrán que ser en un ámbito extraoficial.

8.4. Pruebas del sistema

Estas pruebas se centran en el resultado final obtenido al desarrollar la aplicación por completo, donde es necesario revisar los requisitos funcionales y no funcionales realizados para el capítulo de análisis.

8.4.1. Pruebas Funcionales

Las pruebas funcionales representan un apartado importante en el desarrollo de un aplicativo. En la mayoría de sus casos, corresponde a un costo bajo para el desarrollador y se realiza sobre una funcionalidad específica. En nuestro caso, hemos decidido realizar una prueba funcional sobre el apartado de Control de Salida que, para el cliente final, corresponde al apartado de mayor riesgo de la aplicación puesto que puede dar lugar a errores si la funcionalidad no se ejecuta correctamente.

8.4.2. Pruebas No Funcionales

En este caso, las pruebas no funcionales aplicadas a carga o rendimiento no aplican en el producto ya que la cantidad de usuarios a usarla es unitaria y no paralela ya que la mejora sobre la apertura del portal a los hermanos externos a los administradores del aplicativo no se contempla en la primera versión del mismo aunque para la Universidad de Cádiz quede constancia de que se entrega código fuente del mismo para que se evalúe el conocimiento de realizar una aplicación con diferentes roles.

8.4.3. Pruebas Automáticas

Este tipo de pruebas aportan un importante valor al producto ya que tiene un gran costo de desarrollo y ofrece la posibilidad de tener una base consistente frente a futuros desarrollos. La posibilidad de que el aplicativo crezca y tengamos el respaldo de unas pruebas automatizadas que respalden que el desarrollo nuevo se ha ido realizando sin eliminar ninguna funcionalidad anterior es muy importante de cara a la evolución del aplicativo. Dichas pruebas se denominan de regresión y es la base de nuestras pruebas.

Capítulo 9

Conclusiones y trabajo futuro

En este capítulo se recogen las opiniones y experiencias finales sobre la herramienta, donde se realizará una revisión de los objetivos fijados para la aplicación, un análisis de lo que ha supuesto el proyecto para el autor de este documento, y se incluirán posibles mejoras sobre la herramienta que podrán ser implementadas en el futuro.

9.1. Conclusiones sobre la herramienta

Para este TFG se ha desarrollado una herramienta para la Hermandad del Perdón de Cádiz. Dicha herramienta va a facilitar la gestión de la Hermandad de cara a la salida procesional, a la afiliación de Hermanos y al stock de Túnicas asociadas a la cofradía. Además, se ha previsto mejoras de versiones que tomarán su desarrollo en una fase posterior a la presentación de este trabajo.

Este proyecto ha sido supervisado por el Dr. Carlos Rioja del Río (Departamento de Ingeniería Informática de la Universidad de Cádiz) de forma notable y con un seguimiento bastante acertado.

Revisando los objetivos fijados a obtener para la herramienta final, se concluye lo siguiente:

- La herramienta obtenida es capaz de realizar altas, bajas, edición y listar Personas asociadas a la Hermandad que pueden ser o no, Hermanos de la Cofradía.
- La herramienta obtenida es capaz de realizar altas, bajar, edición y listar Hermanos pertenecientes a la cofradía.
- La herramienta obtenida es capaz de realizar altas, bajar, edición y listar Controles de Salida asociados a la cofradía.
- La herramienta obtenida es capaz de realizar altas, bajar, edición y listar Túnicas pertenecientes a la cofradía.
- La herramienta obtenida posee la facilidad de exportar a PDF o Excel los listados de las diferentes entidades antes mencionadas facilitando así la labor de los gestores de la Hermandad.
- La herramienta obtenida posee diferencias entre perfiles y roles para su uso, según las necesidades acordadas.
- La herramienta obtenida ofrece mejoras significativas en la labor administrativa de la Hermandad ahorrando mucho tiempo de gestión para las personas encargadas de su uso.
- La herramienta obtenida ha cumplido con todos los aspectos necesarios para ser introducida en un servicio web, por el cual, se le puede atribuir un uso continuado a la misma.
- Se ha documentado el proceso de desarrollo de la herramienta, los manuales necesarios y su inclusión en la plataforma Redmine de la Universidad de Cádiz.

- La herramienta obtenida es capaz de enviar de forma automatizada correos a los Hermanos que dispongan de email para las altas o ediciones de sus fichas como Hermanos.
- La herramienta obtenida es capaz de gestionar un pago mediante PayPal en un entorno de pruebas.
- La herramienta obtenida es capaz de exportar un documento PDF como control de salida de forma decorativa a los hermanos que soliciten el control de salida para los años vigentes.
- La herramienta obtenida es actual, vistosa y a la última en tecnología dotando de mas valor al producto entregado.
- La herramienta posee una batería de pruebas automatizadas que ofrecen una garantía sobre el desarrollo futuro que pudiese realizarse.

Por estos motivos, se puede afirmar que la aplicación ha cumplido satisfactoriamente con los objetivos fijados con los usuarios finales, así como, con las expectativas del desarrollador de la misma cumpliendo con el nivel de calidad exigido por este en sus aplicaciones.

9.2. Experiencia personal con el proyecto

Este proyecto, junto con el desarrollado en la asignatura de PINF, ha sido con diferencia el proyecto más amplio y completo en el ámbito académico en el que he trabajado, pues no solo se compone de la programación y los ajustes necesarios para que una aplicación funcione correctamente o de la forma esperada, si no que además, este proyecto tuvo una gran carga de planificación, y aprendizaje tanto para los usuarios finales como para mi en las últimas versiones de ésta tecnología.

Además de una buena planificación, en este proyecto he conocido lo que significa el desarrollo completo de una herramienta, el aprendizaje constante y autodidacta de todo lo que engloba al desarrollo de tus herramientas y el saber aplicar los conocimientos que vas adquiriendo sobre tu trabajo de cada día.

Lo más complicado podría haber sido la labor de compaginar mi jornada laboral con el desarrollo de este proyecto y de tanta dimensión. No se lo recomiendo a nadie.

Pese al estrés por el retraso en la entrega del TFG durante años por trabajar en el sector privado o a la dificultad en la curva de aprendizaje inicial, alinear las versiones de JSF y JPA siendo tarea costosa y que en la Universidad no se habían realizado desarrollo similares en las diferentes asignaturas, esta experiencia me ha aportado grandes valores que seguro me acompañarán a lo largo de toda mi carrera como desarrollador, como una buena planificación, saber trabajar bajo presión, aceptar sugerencias o mejoras sobre las formas en que uno programa, o saber reaccionar frente a los problemas que el código siempre acarrea y confiar en las elecciones que uno toma a la hora de diseñar y desarrollar cualquier aplicación.

Es muy satisfactorio ver como el fruto de tu trabajo puede facilitar enormemente los procesos administrativos del día a día de la Hermandad debido a tu herramienta y aún mas satisfactorio es cuando te indican el buen y gran trabajo que estás haciendo por parte del cliente, tutor, amigos o familiares. Esto te impulsa a seguir dando lo mejor de ti mismo para poder seguir mejorando en tus funciones, o aprendiendo nuevas cosas que ayuden a más personas.

También es importante agradecer al Profesor y Doctor Carlos Rioja del Rio el cual siempre me ha tendido la mano en numerosas ocasiones para poder finiquitar mi trayecto por la Universidad. Gracias por tu atención y enseñanzas. Siempre llevaré conmigo tus clases donde intentabas chocar de frente el mercado laboral con el académico y provocar un shock en tus alumnos haciendo que se den cuenta de lo que tienes en tus manos. ¿Quieres pasar el resto de tu vida arreglando ordenadores?; Dicha pregunta al aire quedó marcada para el resto en muchos alumnos de nuestra generación.

9.3. Trabajo futuro

Aunque los objetivos fijados han sido cumplidos correctamente, siempre se puede mejorar el estado y las funcionalidades de toda herramienta desarrollada, por ello, en este apartado se indican ciertas acciones que se podrían implementar en la herramienta con el fin de conseguir mayor robustez en la misma.

9.3.1. Mejoras

Manejo de sesión y paralelismo de Usuarios

La herramienta en la versión de entrega del TFG no está pensada para el acceso masivo de usuarios por lo que, para la próxima versión, se podría plantear el desarrollo y control de sesiones mediante cookies para controlar el acceso masivo de personas afiliadas a la Hermandad que necesiten renovar su control de salida.

Pagos en entornos productivos

El pago en esta versión se ha desarrollado con PayPal en un entorno de pruebas facilitado por la propia compañía para probar su REST, pero por falta de datos bancarios con la Hermandad no hemos podido facilitar un pago productivo y real que determine la verificación del método.

Implementación y migración a las nuevas versiones de JSF y PrimeFaces

Implementar un aplicativo web que junte JPA y JSF puede llegar a ser tedioso por el conflicto de versiones y no estamos contando con el framework de PrimeFaces. Desde que se empezó el proyecto hasta que se ha terminado su desarrollo se han publicado cuatro versiones diferentes de la librería lo cual hace que la migración del producto deba ser constante y concisa ya que la labor de los componentes PrimeFaces facilita el desarrollo del aplicativo.

Automatización de procesos para cada año

La aplicación por naturaleza tiene un ciclo de vida constante en el tiempo y existen parametrizaciones en la base de datos que están fijadas al año, como pueden ser los recibos de hermano o de control de salida. Estos datos habría que incluirlos año a año para que la aplicación siga siendo útil.

Limpieza de datos históricos en Base de Datos

Al ser una aplicación constante en el tiempo, será necesario almacenar el histórico anual en tablas alternativas para poder tener una lectura limpia y real de las tablas maestras en la aplicación.

Administración del Patrimonio

La aplicación no cuenta con el control y mantenimiento del patrimonio que sacan a la calle así que en una próxima versión se estudiará dotar a la aplicación de un menú donde se puedan realizar dichas labores.

Realización de una APP Android o iOS

Por petición expresa del cliente y adaptación a los nuevos tiempos, se quiere desarrollar una APP para la Hermandad y se estudiará para qué plataforma y mercado se haría en un primer momento.

Apéndice A

Manual de instalación

Para la puesta en marcha de la herramienta y el buen uso de esta, será necesario el uso de las siguientes herramientas:

- Java
- WAMP
- Eclipse
- Git

Este manual indicará cómo instalar dichas herramientas e interfaces. Una vez instalado las aplicaciones descritas será posible utilizar la herramienta correctamente.

A.1. Java

Se recomienda, antes de proceder con la instalación en línea, desactivar el cortafuegos de Internet. En algunos casos, la configuración del cortafuegos predeterminado se establece para rechazar todas las instalaciones automáticas o en línea, como la instalación en línea de Java. Si el cortafuegos no se configura correctamente, podría impedir la operación de descarga/instalación de Java en determinadas circunstancias. Consulte las instrucciones del manual específico del cortafuegos de Internet para desactivarlo.

Para la descarga de Java es necesario que acceda a la siguiente página o a la siguiente URL <https://www.java.com/es/download/manual.jsp> y seguir las instrucciones que indica el distribuidor en función a su versión de sistema operativo y arquitectura.

Una vez descargado y ejecutado el instalador obtenemos una imagen como la siguiente y debemos seguir la guía de instalación.

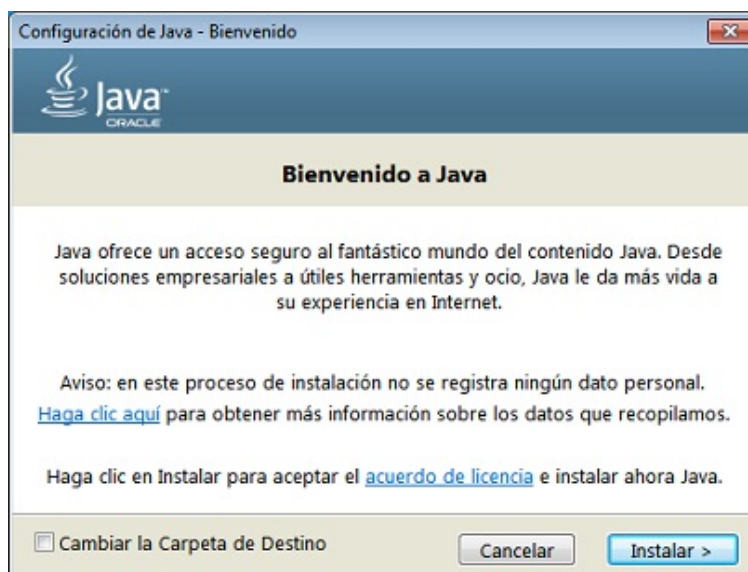


Figura A.1: Bienvenido a Java

A.2. WAMP

Lo primero que debemos hacer es descargar el programa desde su página web principal. El programa es totalmente gratuito y está libre de software publicitario y no deseado. En su página web o a la siguiente URL <http://www.wampserver.com/en/> vamos a poder elegir si queremos descargar la versión de 32 bits o la de 64 bits. En nuestro caso vamos a trabajar con la versión de 64 bits, ya que tanto nuestro procesador como nuestro sistema operativo lo permiten. Pulsamos sobre la versión deseada de Wamp Server y, en la ventana que nos aparece, pulsamos sobre el enlace *download directly*. Seguimos el asistente de instalación que hemos descargado y no deberíamos tener problemas de instalación. Una vez realizado esto, ya tenemos un mySQL y Apache listo para poder ser ejecutado en nuestro localhost.

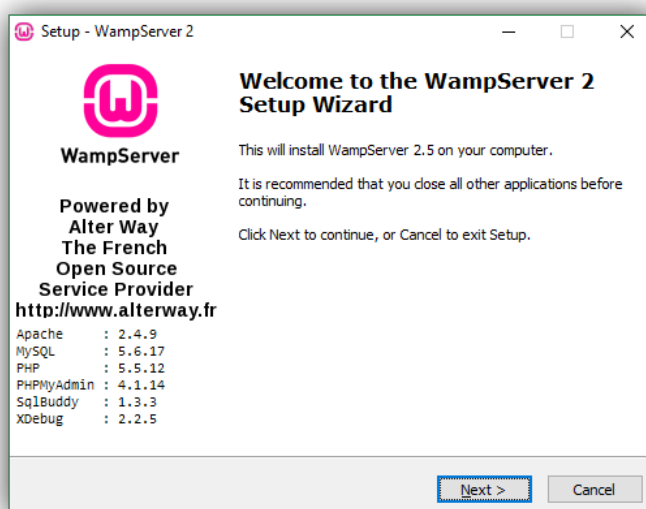


Figura A.2: Instalación de WAMP

Para la instalación de las tablas y para que nuestro documento no sea muy repetitivo en mostrar código elemental, junto a la entrega de este documento se entregará en la subida del código fuente

una carpeta con los scripts a ejecutar.

A.3. Eclipse

La instalación de Eclipse es muy sencilla, se realiza desde su página web o a la siguiente URL <https://eclipse.org/downloads/eclipse-packages/>. Debemos seleccionar la última versión para nuestro sistema operativo y descargar su instalador asociado.

Cuando termine la descarga empezará la instalación. Si no lo hace debemos ir a la carpeta donde lo hemos descargado y ejecutar el archivo con doble click. Al comenzar la instalación, nos sale una advertencia de seguridad que debemos aceptar haciendo click en *Ejecutar*. Podremos elegir si descargar el entorno para Java Standar, Java EE y otras muchas versiones más específicas a las que siempre podremos ampliar instalando nuevos paquetes desde nuestro IDE. En esta ocasión vamos a instalar *Eclipse IDE for Java Developers (Versión para desarrolladores Java)*.

Antes de arrancar el programa debemos seleccionar el espacio de trabajo. Eclipse guarda todos nuestros proyectos en el espacio de trabajo. Podemos dejarlo como está o poner la ruta que queramos. Cada vez que arranque eclipse nos preguntará esto, a menos que marquemos la casilla *Use this as the default and do not ask again*. Luego hacemos click en *OK*.

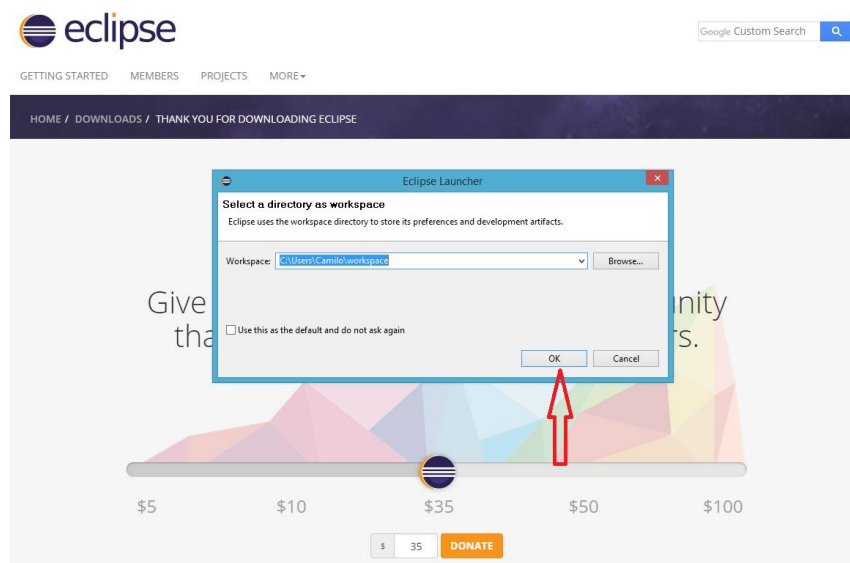


Figura A.3: Nuestro IDE: Eclipse

Ya nos encontramos ante la interfaz de eclipse. Lo primero que se abre es la pantalla de bienvenida, en la que tenemos varias opciones, tutoriales y recomendaciones básicas.

A.4. Git

Para empezar, entra a la página de Git o a la siguiente URL <https://git-scm.com/> y darle haz click sobre el sistema operativo que estás trabajando. Dicha acción genera una descarga de un archivo el cual debemos ejecutar.

Como muchos de los instaladores en Windows, debemos de aceptar las opciones por defecto y darle Next (siguiente) a todo hasta que nos salga el botón de instalar. Si seguimos estos pasos, instalaremos Git con sus opciones básicas que será suficiente para nuestro trabajo Y ésta será la terminal que usaremos para ingresar comandos Unix y de Git sobre todo:

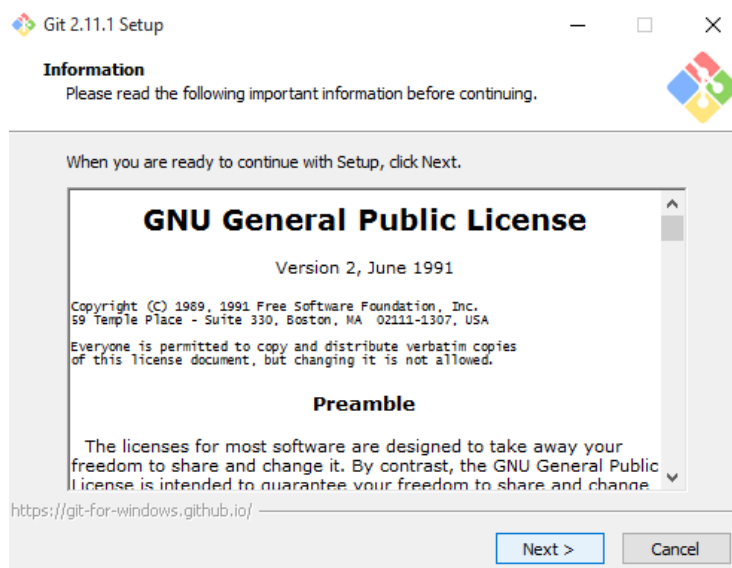


Figura A.4: Un controlador de versiones... GIT

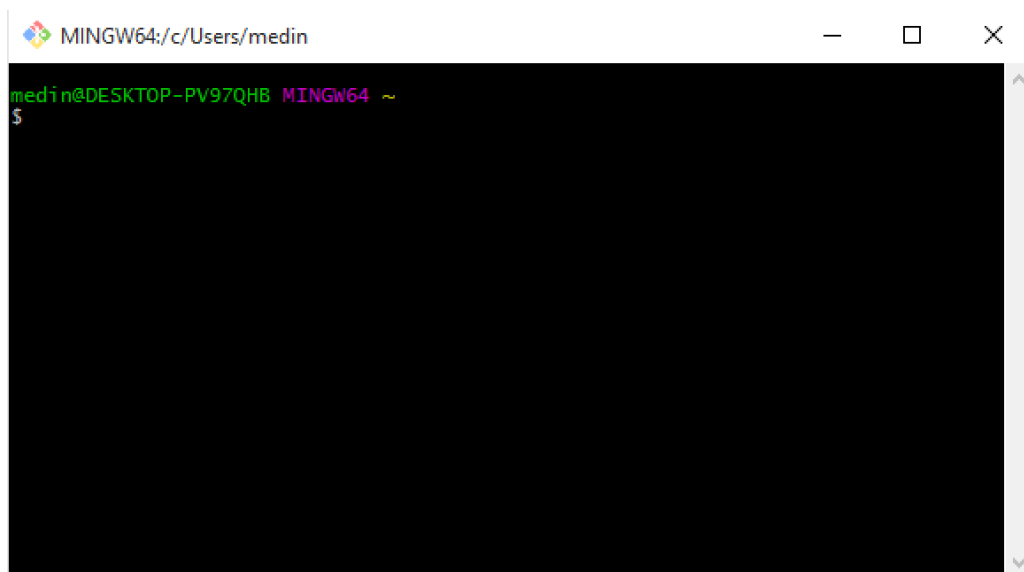


Figura A.5: Bash de GIT

A.5. Configuración de la herramienta

Una vez realizada la instalación de todas las herramientas necesarias para el uso de nuestra aplicación, se describe a continuación las configuraciones que debe realizar para obtener la herramienta funcional y de forma correcta. Para ello, debemos dirigirnos al GitHub del desarrollador y hacer un clone o descarga del proyecto. Cuando esto haya acabado, abriremos Eclipse y procederemos a importar el proyecto que hemos descargado.

Lo siguiente que debemos hacer es añadir un servidor de aplicaciones. En nuestro caso, por eclipse hemos usado Apache TomCat y podemos hacerlo desde la pestaña de Servidores, click con el botón derecho en añadir, y añadimos nuestro TomCat que creamos conveniente para nuestro proyecto. En nuestro caso un TomCat 9.0.

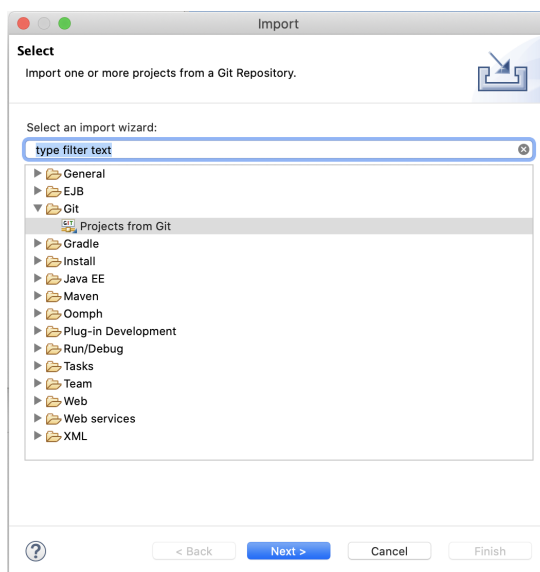


Figura A.6: Importamos nuestro proyecto

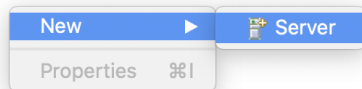


Figura A.7: Añadimos Apache TomCat

Tras esta configuración, podemos arrancar el servidor de aplicaciones con nuestro XAMP/WAMP corriendo y acceder a las direcciones de localhost que faciliten dichos servicios. En nuestro caso, tenemos **<http://localhost:8888/phpMyAdmin/>** para el servicio de base de datos y **<http://localhost:8080/cofrad>** para el acceso a la aplicación.

Apéndice B

Manual de desarrollador

Una vez realizada la instalación de todas las herramientas necesarias para el uso de nuestra aplicación y su configuración, se indica a continuación la composición del proyecto, para exponer como tratar el código en el mismo, donde realizar ampliaciones con nuevas funcionalidades o donde se encuentran cada módulo de la herramienta.

B.1. Estructura del código

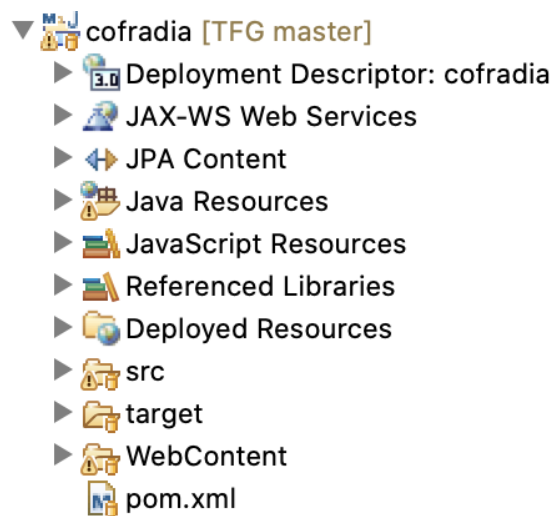


Figura B.1: Estructura del proyecto.

Como puede apreciar en la anterior figura, la herramienta se compone de diversos apartados definido por una arquitectura personalizada al desarrollo del producto. Encontramos un archivo **pom.xml** donde estará definidas todas las dependencias maven del proyecto. Por otro lado, tenemos dos bloques de desarrollo:

- **Java Resources** En este apartado tendremos ubicado todo el desarrollo backend del producto y sus test asociados.
- **WebContent** En este apartado tendremos ubicado todo el desarrollo front del producto.

Debido a las diferencias que existen entre cada directorio, se desglosarán a continuación directorio a directorio su contenido.

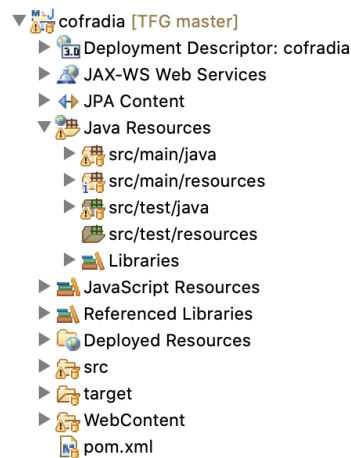


Figura B.2: Estructura Java del proyecto.

B.1.1. Java Resources

Dentro de este apartado, podemos encontrar las clases Java y archivos de configuración de nuestro proyecto como pueden ser el *persistence.xml*, *setting.xml* en caso de que fuese necesario.. además las propias clases Java diferenciadas por funcionalidad y marco de desarrollo.

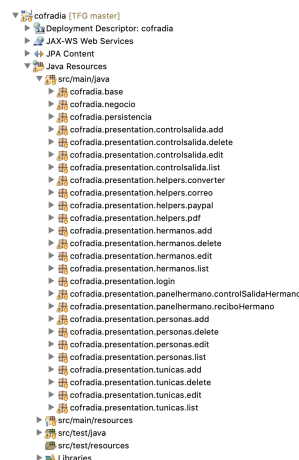


Figura B.3: Arquitectura de paquetado Java.

B.1.2. Base

En este apartado obtenemos el código base diseñado para nuestro producto. En este caso, aquí tendremos organizados las constantes utilizadas en la aplicación, funcionalidades útiles que se tengan en común en el mismo, validaciones, y configuración necesaria para el conector de Eclipse con MySQL (*DataBaseBuilder.java*).

Además, en este directorio encontramos la clase **Presentacion.java**, en dicha clase, podemos comprobar que existe una relación con el servicio de mensajes que muestra la aplicación ordenado por eventos: error, labels, mensajes de la aplicación... Será en dicha referencia donde debemos actuar si queremos añadir nuevos mensajes o editar los ya existentes. Hay que tener en cuenta que todo el desarrollo incluido en el paquete *Base* es esencial para nuestra aplicación ya que desde dicha columna se ha construido todo el producto restante.



Figura B.4: Estructura Java: Base.

B.1.3. Persistencia

En este apartado obtenemos las entidades que representamos de nuestra base de datos. Dichas entidades se han definido con el cliente y creado en nuestra base de datos. Para poder verlas en nuestro aplicativo Java es necesario configurar el archivo persistence.xml de la siguiente forma:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <-persistence xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://
   java.sun.com/xml/ns/persistence/persistence_2_0.xsd" xmlns:xsi="http://
   www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/
   persistence" version="2.0">
4
5
6  <-persistence-unit transaction-type="JTA" name="cofradia">
7
8  <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
9
10 <class>cofradia.persistencia.Hermano</class>
11
12 <class>cofradia.persistencia.ControlSalida</class>
13
14 <class>cofradia.persistencia.Paso</class>
15
16 <class>cofradia.persistencia.Persona</class>
17
18 <class>cofradia.persistencia.Procesion</class>
19
20 <class>cofradia.persistencia.Recibo</class>
21
22 <class>cofradia.persistencia.TipoPago</class>
23
24 <class>cofradia.persistencia.TipoRecibo</class>
25
26 <class>cofradia.persistencia.Tunica</class>
27
28 <class>cofradia.persistencia.Usuario</class>
29
30 <exclude-unlisted-classes>>false</exclude-unlisted-classes>
31
32

```

```

33  -<properties>
34
35  <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
36
37  <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:8889/
    COFRADIA"/>
38
39  <property name="javax.persistence.jdbc.user" value="root"/>
40
41  <property name="javax.persistence.jdbc.password" value="root"/>
42
43  <property name="eclipselink.weaving" value="false"/>
44
45  <property name="eclipselink.allow-zero-id" value="true"/>
46
47  </properties>
48
49  </persistence-unit>
50
51  </persistence>

```

Código B.1: Persistence.xml

Podemos comprobar que en dicho XML, configuramos nuestra conexión JDBC con el Driver de MySQL. Dicha configuración permite la conexión desde Eclipse a nuestra base de datos. Por último, en la propiedad `<class>` se definen las entidades.

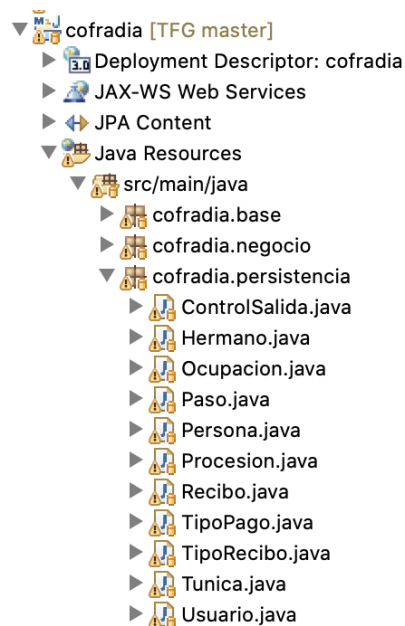


Figura B.5: Estructura Java: Persistencia.

Dentro de cada clase Java de la persistencia, debemos definir el esquema y nombre de las columnas además de definir los tipos de datos que se van a usar. Se adjunta una clase de ejemplo:

```

1  package cofradia.persistencia;
2
3  import java.util.Date;
4
5  import javax.persistence.Column;
6  import javax.persistence.Entity;
7  import javax.persistence.GeneratedValue;
8  import javax.persistence.GenerationType;
9  import javax.persistence.Id;
10 import javax.persistence.JoinColumn;
11 import javax.persistence.OneToOne;

```

```

12 import javax.persistence.Table;
13 import javax.validation.constraints.NotNull;
14
15 import cofradia.base.Numeros;
16
17 @Entity(name = Hermano.ENTITY_NAME)
18 @Table(name = Hermano.TABLE_NAME)
19 public class Hermano {
20     /**
21      *
22      */
23     private static final long serialVersionUID = -762872045574496352L;
24     public static final String ENTITY_NAME = "Hermano";
25     public static final String TABLE_NAME = "HERMANO";
26     public static final String COL_NAT_ID_HERMANO = "ID_HERMANO";
27     public static final String COL_NAT_NUMERO = "NUMERO";
28     public static final String COL_NAT_FECHA_REGISTRO = "FECHA_REGISTRO";
29     public static final String COL_NAT_ID_PERSONA = "ID_PERSONA";
30     public static final String COL_NAT_ID_TUNICA = "ID_TUNICA";
31     public static final String COL_NAT_ID_PERSONA_AUTORIZADA = "ID_PERSONA_AUTORIZADA"
32     ;
33
34     @Id
35     @NotNull
36     @GeneratedValue(strategy = GenerationType.IDENTITY)
37     @Column(name = Hermano.COL_NAT_ID_HERMANO, nullable = false, precision =
38         Numeros.I_12, scale = Numeros.I_0)
39     private Long id;
40
41     @NotNull
42     @Column(name = Hermano.COL_NAT_NUMERO)
43     private int numero;
44
45     @NotNull
46     @Column(name = Hermano.COL_NAT_FECHA_REGISTRO)
47     private Date fechaRegistro;
48
49     @OneToOne
50     @NotNull
51     @JoinColumn(name = Hermano.COL_NAT_ID_PERSONA, referencedColumnName =
52         Persona.COL_NAT_ID_PERSONA)
53     private Persona persona;
54
55     @OneToOne
56     @JoinColumn(name = Hermano.COL_NAT_ID_TUNICA, referencedColumnName =
57         Tunica.COL_NAT_ID_TUNICA)
58     private Tunica tunica;
59
60     @OneToOne
61     @JoinColumn(name = Hermano.COL_NAT_ID_PERSONA_AUTORIZADA, referencedColumnName =
62         Persona.COL_NAT_ID_PERSONA)
63     private Persona personaAutorizada;
64
65     public Long getId() {
66         return id;
67     }
68
69     public int getNumero() {
70         return numero;
71     }
72
73     public void setNumero(int numero) {
74         this.numero = numero;
75     }
76
77     public Date getFechaRegistro() {
78         return fechaRegistro;
79     }
80
81     public void setFechaRegistro(Date fechaRegistro) {

```

```
77     this.fechaRegistro = fechaRegistro;
78 }
79
80 public Persona getPersona() {
81     return persona;
82 }
83
84 public void setPersona(Persona persona) {
85     this.persona = persona;
86 }
87
88 public Tunica getTunica() {
89     return tunica;
90 }
91
92 public void setTunica(Tunica tunica) {
93     this.tunica = tunica;
94 }
95
96 public void setId(Long id) {
97     this.id = id;
98 }
99
100 public Persona getPersonaAutorizada() {
101     return personaAutorizada;
102 }
103
104 public void setPersonaAutorizada(Persona personaAutorizada) {
105     this.personaAutorizada = personaAutorizada;
106 }
107
108 @Override
109 public int hashCode() {
110     final int prime = 31;
111     int result = 1;
112     result = prime * result + ((this.id == null) ? 0 : this.id.hashCode());
113     return result;
114 }
115
116 /*
117  * (non-Javadoc)
118  * @see java.lang.Object#equals(java.lang.Object)
119  */
120 @Override
121 public boolean equals(Object obj) {
122     if (this == obj) {
123         return true;
124     }
125     if (obj == null) {
126         return false;
127     }
128     if (this.getClass() != obj.getClass()) {
129         return false;
130     }
131     Hermano other = (Hermano) obj;
132     if (this.id == null) {
133         if (other.id != null) {
134             return false;
135         }
136     } else if (!this.id.equals(other.id)) {
137         return false;
138     }
139     return true;
140 }
141
142 }
```

Código B.2: Hermano.java

B.1.4. Negocio

Igual que tenemos la persistencia, debemos definir el negocio de la aplicación. En otros ámbitos y filosofías de desarrollo estaríamos hablando de los DAO, que es como mas comunmente se conocen a este tipo de clases.



En estas clases, está representado toda consulta necesaria para nuestra aplicación y será en este paquete dónde debemos crear nuevas clases que necesiten de una consulta en base de datos para nuestro uso.

```

1 package cofradia.negocio;
2
3 import java.sql.PreparedStatement;
4 import java.sql.ResultSet;
5 import java.sql.SQLException;
6 import java.util.ArrayList;
7 import java.util.List;
8
9 import javax.ejb.Stateless;
10
11 import cofradia.base.DataBaseBuilder;
12 import cofradia.persistencia.Hermano;
13 import cofradia.persistencia.Paso;
14
15 @Stateless
16 public class Pasos extends DataBaseBuilder {
17
18     public Pasos() {
19         super();
20     }
21
22     public boolean findPasoByHermano(Hermano hermano) {
23         boolean var = false;
24         try {
25             PreparedStatement paso = db.prepareStatement("SELECT * FROM COFRADIA.PASO
26                 WHERE ID_PASO = ?");
27             paso.setString(1, String.valueOf(hermano.getTunica().getPaso().getId()));
28             ResultSet resultPaso = paso.executeQuery();
29             while (resultPaso.next()) {
30                 var = true;
31                 hermano.getTunica().setPaso(new Paso());
32                 hermano.getTunica().getPaso().setId(Long.valueOf(resultPaso.getString(
33                     Paso.COL_NAT_ID_PASO)));

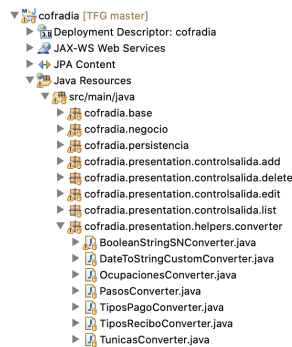
```

```
32         hermano.getTunica().getPaso().setDescription(resultPaso.getString(
33             Paso.COL_NAT_DESCRIPCION));
34     } catch (SQLException e) {
35         e.printStackTrace();
36     }
37     return var;
38 }
39
40 public List<Paso> findAll() {
41     List<Paso> list = new ArrayList<Paso>();
42     try {
43         PreparedStatement paso = db.prepareStatement("SELECT * FROM COFRADIA.PASO");
44         ResultSet resultPaso = paso.executeQuery();
45         while (resultPaso.next()) {
46             Paso obj = new Paso();
47             obj.setId(Long.valueOf(resultPaso.getString(Paso.COL_NAT_ID_PASO)));
48             obj.setDescription(resultPaso.getString(Paso.COL_NAT_DESCRIPCION));
49             list.add(obj);
50         }
51     } catch (SQLException e) {
52         e.printStackTrace();
53     }
54     return list;
55 }
56
57 public Paso findById(Long tipo) {
58     Paso obj = null;
59     try {
60         PreparedStatement paso = db.prepareStatement("SELECT * FROM COFRADIA.PASO
61             WHERE ID_PASO = ?");
62         paso.setString(1, String.valueOf(tipo));
63         ResultSet resultPaso = paso.executeQuery();
64         while (resultPaso.next()) {
65             obj = new Paso();
66             obj.setId(Long.valueOf(resultPaso.getString(Paso.COL_NAT_ID_PASO)));
67             obj.setDescription(resultPaso.getString(Paso.COL_NAT_DESCRIPCION));
68         }
69     } catch (SQLException e) {
70         e.printStackTrace();
71     }
72     return obj;
73 }
74
75 public Paso findByDescripcion(String tipo) {
76     Paso obj = null;
77     try {
78         PreparedStatement paso = db.prepareStatement("SELECT * FROM COFRADIA.PASO
79             WHERE DESCRIPCION = ?");
80         paso.setString(1, tipo);
81         ResultSet resultPaso = paso.executeQuery();
82         while (resultPaso.next()) {
83             obj = new Paso();
84             obj.setId(Long.valueOf(resultPaso.getString(Paso.COL_NAT_ID_PASO)));
85             obj.setDescription(resultPaso.getString(Paso.COL_NAT_DESCRIPCION));
86         }
87     } catch (SQLException e) {
88         e.printStackTrace();
89     }
90     return obj;
91 }
92 }
```

Código B.3: Pasos.java

B.1.5. Helpers

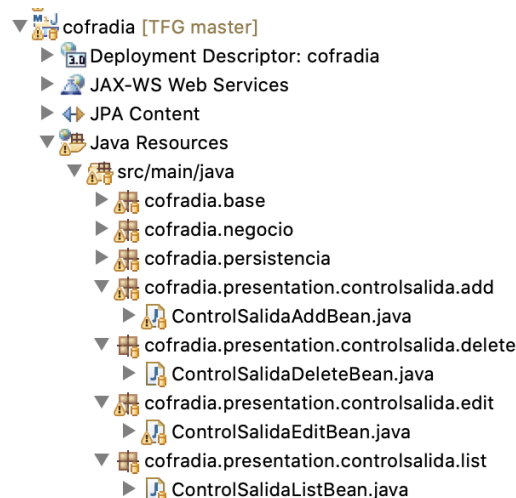
En esta sección hablaremos de los helpers, converters, y otras clases de utilidad que tenemos en nuestro proyecto. En función a la arquitectura, debemos diferenciar el empaquetado para evitar mezclar funcionalidades entre los empaquetados.



B.1.6. Controladores

En esta sección hablaremos de los beans. Los beans son necesarios en nuestro aplicativo ya que hacen comunicación entre el front y el back mediante nuestras páginas JSF. En otros modelos de desarrollo este concepto se definiría como *Controlador*. En estas clases están definidas las funcionalidades que deberá tener la sección en concreto que nos encontramos.

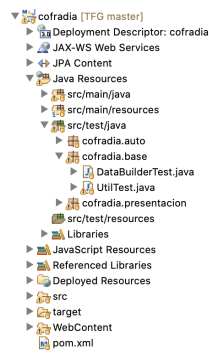
Se han dividido por funcionalidad y módulo para evitar mezclar código y sobrecargar la misma clase de funcionalidades que no le pertenecen. Esto hace que los desarrolladores que quieran proseguir con la tarea tengan un camino limpio sobre el que desarrollar sin investigar o indagar en nuestro código mas de lo debido.



B.1.7. Test

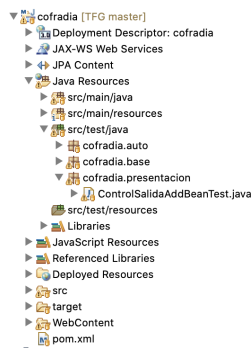
En el apartado de test tenemos dos grupos principales: Test automatizados, y test unitarios. Para ello, es necesario crear una arquitectura de Test base. Dicha arquitectura la vemos reflejada en la siguiente imagen:

Si necesitamos implementar funcionalidades que tengan que convivir en ambos tipos de test, desarrollaremos la necesidad en este paquete.



B.1.7.1. Unitarios

Estos test existen para satisfacer las pruebas unitarias de nuestro aplicativo. Dentro de este apartado, vamos a ir introduciendo test organizados con la misma arquitectura que seguimos para todo el proyecto.



B.1.7.2. Automatizados

Estos test existen para satisfacer las pruebas automatizadas de nuestro aplicativo. Dentro de este apartado, vamos a ir introduciendo test organizados con la misma arquitectura que seguimos para todo el proyecto.



Apéndice C

Manual de usuario

Se incluye a continuación el manual de usuario asociado a la herramienta. Dicho manual está desarrollado en su totalidad por el autor de este trabajo y contempla las instrucciones de uso de la herramienta para el usuario final.

C.1. Introducción

El presente documento es un manual esquemático de la aplicación para los usuarios finales. Debido a la cantidad diferente de hermanos que hacen vida en la hermandad es necesario realizar un manual que ayude a la gestión de datos de los hermanos. Estos datos contribuyen a un proceso de mejora y agilización del modelo tradicional del que se venía trabajando, logrando unos ahorros en tiempo, claridad en las acciones, y redundancia de datos para los que trabaja la Hermandad.

C.2. Inicio

Al entrar en el aplicativo nos encontramos en la pantalla de inicio. Dicha pantalla nos ofrece una información clara y concisa de lo que necesita la Hermandad y las funcionalidades de la herramienta. Para poder hacer uso de la herramienta, es necesario hacer click en Entrar.

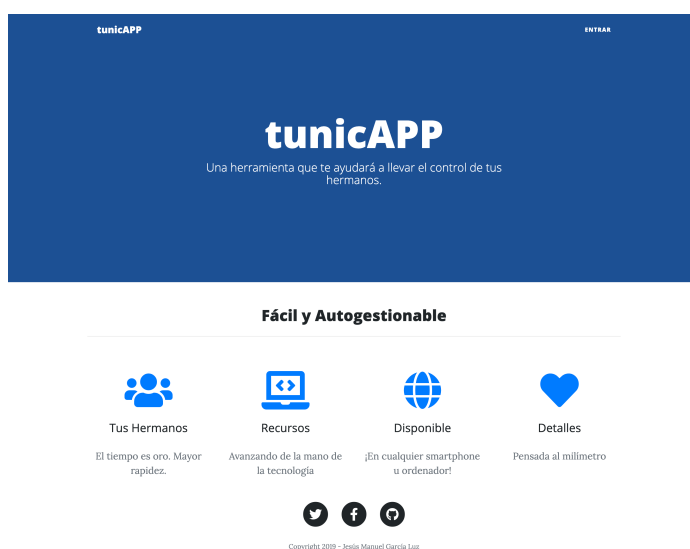


Figura C.1: Pantalla de inicio.

C.3. Login

En la pantalla de Login tenemos un formulario donde introduciremos el usuario y contraseña facilitado por TUNICAPP ya que, el administrador al darnos de alta, la herramienta te envía al correo las claves de acceso al sistema y es el personal encargado del mantenimiento de los datos y la fiabilidad del entorno productivo el que realiza la modificación del rol del usuario. De esta forma, nos aseguramos que existen al menos dos personas independientes del conocimiento del rol de un usuario.

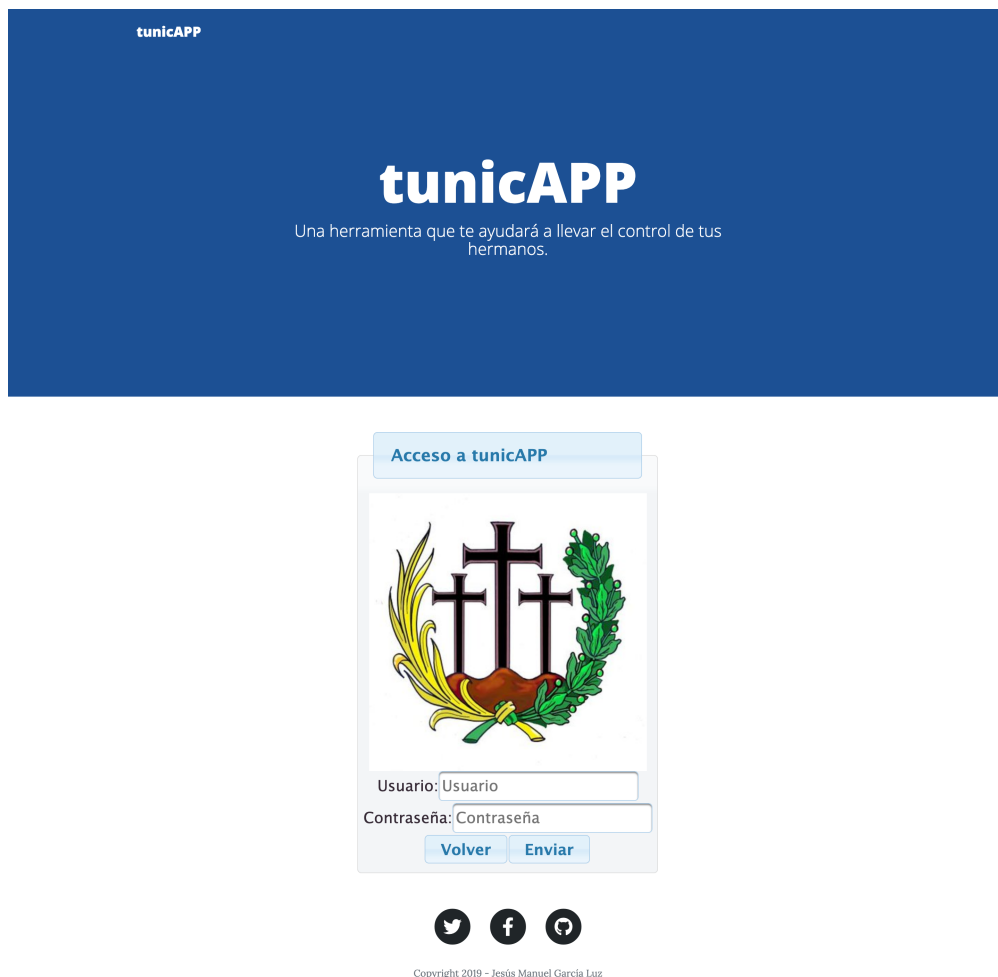


Figura C.2: Pantalla de Login.

C.4. Pantalla principal Administrador

Una vez introducido el usuario y la contraseña en la pantalla de Login, obtenemos la pantalla de Administrador en el caso de que el rol introducido fuese de administración. En dicha pantalla podemos visualizar información sobre la aplicación y sus diferentes opciones situadas en el panel superior de la pantalla. En función con la funcionalidad que queramos trabajar tenemos representado:

- Personas
- Control de Salida
- Hermanos
- Túnicas

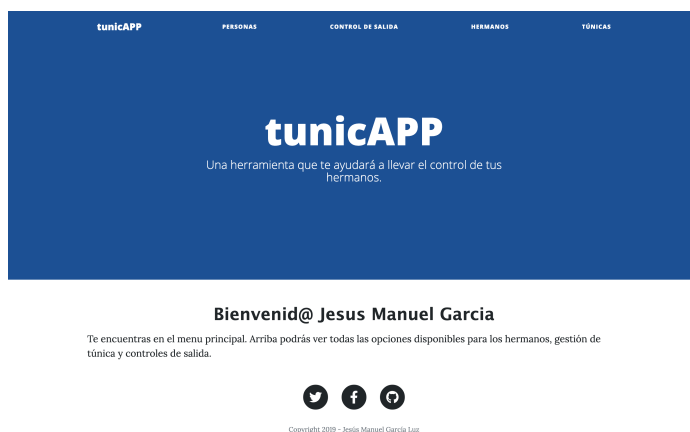


Figura C.3: Pantalla de Inicio para rol Administrador.

C.5. Personas

En dicho panel podemos encontrar funcionalidades asociadas a las personas de nuestro sistema dentro Hermandad. Cada funcionalidad será desarrollado en subsecciones de este mismo apartado.

C.5.1. Listado de Personas

En esta pantalla, podemos ver datos principales y de un primer vistazo de las personas dadas de alta en el sistema. Esta forma de representar la información es importante para poder obtener de un vistazo rápido atributos que hemos considerado esenciales entre el cliente y el desarrollador como puede ser DNI, Nombre, Teléfono o dirección física.

Queremos obtener esta información casi al instante para poder realizar nuestra labor de gestión de la hermandad de forma ágil y correcta.

Además, el sistema de listado se encuentra paginado dando lugar a una configuración mas personal en función a la persona que lo esté usando en ese momento. Podemos ver 5 registros, 10, 20.. dependiendo del usuario dando como resultado una información paginada.

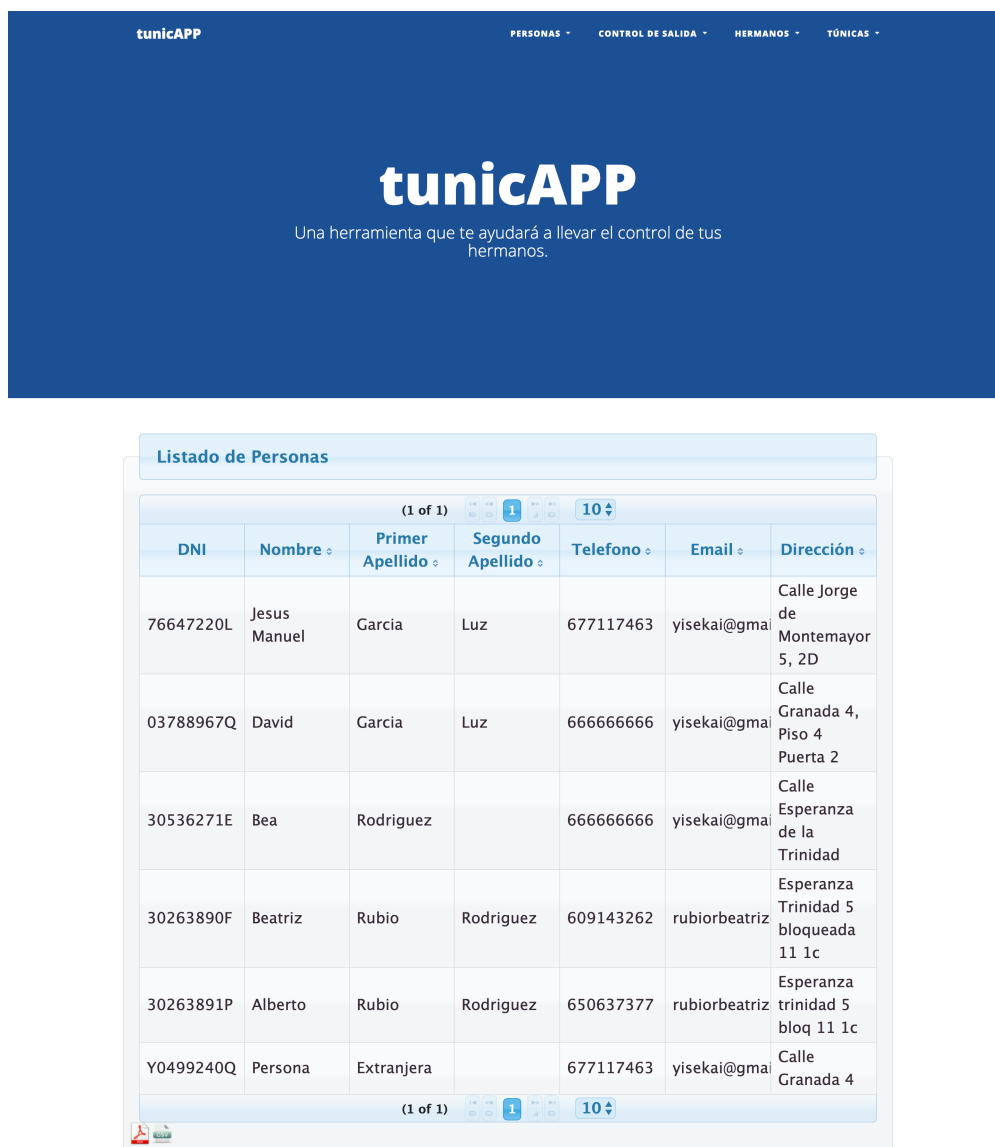


Figura C.4: Pantalla de Listado de Personas.

C.5.2. Añadir Persona

En esta pantalla podemos añadir una persona al sistema. En dicho formulario se reflejan los campos obligatorios con el asterisco (*) y, en caso de realizar alguna acción incorrecta o una inconsistencia con los datos, el sistema te lo hará saber mediante un error con el campo asociado.

Además, en la página se encuentra la posibilidad de aceptar el tratamiento de los datos por parte de las personas que se le solicitará a la persona que se da de alta la aceptación de los mismos.



Alta

Formulario

DNI/NIE (*)	Número de Documento (*)	
<input type="text" value="Seleccione"/>	<input type="text"/>	
Nombre (*)	Primer Apellido (*)	
<input type="text"/>	<input type="text"/>	
Segundo Apellido	Teléfono (*)	Email
<input type="text"/>	<input type="text"/>	<input type="text"/>
Fecha de Nacimiento (*)		
<input type="text" value="dia/mes/año"/>		
Domicilio (*)		
<input type="checkbox"/>		
Autorizo a la Venerable Cofradía de Penitencia del Santísimo Cristo del Perdón y María Santísima del Rosario en sus Misterios Dolorosos al tratamiento de mis datos personales (incluyendo su incorporación en los correspondientes ficheros) incorporados en la solicitud y en otros documentos normalizados relativos a la acción formativa, tal y como se dispone en la Ley Orgánica 15/1999, de 13 de diciembre de Protección de Datos de Carácter Personal (LOPD), para su tratamiento o consulta en las acciones que dicha corporación vea oportuna.		
<input type="button" value="Enviar"/>		



Copyright 2019 - Jesús Manuel García Luz

Figura C.5: Pantalla para añadir Personas.

C.5.3. Editar Personas

En este apartado, podemos editar la información facilitada de la persona. La pantalla ofrece un formulario cumplimentado con los datos de la persona y nos permite modificar dicha información. Al finalizar nuestra tarea, podemos darle click a Editar y el sistema almacenará de forma automática toda la información introducida previo a validaciones.

En caso de éxito, el sistema notificará mediante mensaje de pantalla un mensaje de confirmación del guardado, y en caso negativo, el sistema ofrecerá mediante pantalla un mensaje con el motivo del error.

C.5.4. Eliminar Persona

En este apartado, podemos eliminar del sistema a una persona. Para ello, será necesario cumplimentar el apartado de DNI y pulsar en Borrar. En caso positivo, el sistema notificará mediante pantalla que el borrado ha sido con éxito.

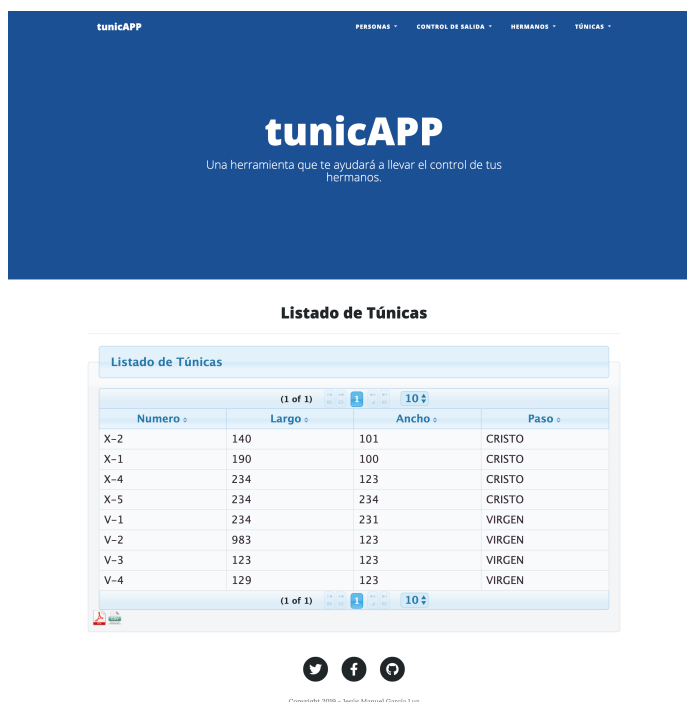
En caso de error, el sistema notificará el mensaje de error y podrás intentarlo de nuevo como cualquier otra funcionalidad.

C.6. Túnicas

En dicho panel podemos encontrar funcionalidades asociadas a las túnicas de nuestro sistema dentro Hermandad. Cada funcionalidad será desarrollado en subsecciones de este mismo apartado.

C.6.1. Listado de túnicas

En este apartado, podemos consultar las túnicas existentes en la Hermandad. Además, de un primer vistazo, obtenemos información del número, largo y ancho, y titular al que pertenece la túnica. Se ofrece la posibilidad de exportar la información a Excel y a PDF para poder compartir información con personal externo o realizar métricas/gráficas de la información del sistema. En dicho listado también tenemos la oportunidad de paginar la información por número de registros.



tunicAPP

Una herramienta que te ayudará a llevar el control de tus hermanos.

Listado de Túnicas

Numero	Largo	Ancho	Paso
X-2	140	101	CRISTO
X-1	190	100	CRISTO
X-4	234	123	CRISTO
X-5	234	234	CRISTO
V-1	234	231	VIRGEN
V-2	983	123	VIRGEN
V-3	123	123	VIRGEN
V-4	129	123	VIRGEN

Copyright 2019 - Jesús Manuel García Luz

Figura C.6: Pantalla de Listado de Túnicas.

C.6.2. Añadir Túnica

En este apartado el sistema nos brinda la oportunidad de dar de alta una túnica en el sistema. El sistema nos ofrecerá un formulario con una serie de campos que el usuario debe cumplimentar.

En este apartado se puede añadir túnicas ya existentes previo al sistema y túnicas nuevas. En el caso de añadir una túnica previa al sistema se deberá especificar el número de la túnica con el formato Titular-Número. Ejemplo: V-21. Esto hace que el inventario que ya teníamos previo al

sistema lo podamos incorporar a la aplicación. Por otro lado, si necesitamos añadir túnicas nuevas, basta con seleccionar el titular al que pertenece la túnica (Cristo o Virgen en el seleccionable de Paso) y las dimensiones de la misma.

Esto hará que el sistema haga un recuento de las existentes de ese titular y añada una mas a la última numeración. Se ofrece la posibilidad de exportar la información a Excel y a PDF para poder compartir información con personal externo o realizar métricas/gráficas de la información del sistema. En dicho listado también tenemos la oportunidad de paginar la información por número de registros.

tunicAPP PERSONAS CONTROL DE SALIDA HERMANOS TÚNICAS

tunicAPP
Una herramienta que te ayudará a llevar el control de tus hermanos.

Añadir Túnica

Formulario

Número

Paso (*)
Seleccione:

Largo de la túnica (cm) (*)

Ancho de la túnica (cm) (*)

Enviar Limpiar

Twitter Facebook Messenger

Copyright 2019 - Jesús Manuel García Lora

Figura C.7: Pantalla para añadir Túnicas.

C.6.3. Editar Túnica

En esta pantalla, podemos editar una túnica que previamente hemos decidido buscar. Para ello, en la pantalla de Editar Túnica, tenemos un formulario donde debemos introducir la numeración de la túnica en el formato Titular-Número. Ejemplo: X-1. El número debe ser exacto al existente, sino, el sistema mostrará por pantalla un error de búsqueda. Una vez hayamos encontrado la túnica para poder editarla, el sistema ofrecerá como información no modificable los campos de número y paso perteneciente. Dicha información no puede modificarse. Sin embargo, el sistema nos ofrece la oportunidad de editar información relevante a la dimensión del mismo como son el largo y ancho de la túnica.

tunicAPP

PERSONAS CONTROL DE SALIDA HERMANOS TÚNICAS

tunicAPP

Una herramienta que te ayudará a llevar el control de tus hermanos.

Editar Túnica

Panel de Búsqueda

Número
X-2

Buscar Limpiar

Datos de la túnica X-2:

Número X-2	Paso CRISTO
Largo de la túnica (cm) 140	Ancho de la túnica (cm) 101

Editar

Twitter Facebook YouTube

Copyright 2019 - Jesús Manuel García Lina

Figura C.8: Pantalla para editar la túnica.

C.6.4. Eliminar túnica

En dicha funcionalidad, debemos introducir el número exacto de la túnica a eliminar y hacer click en Borrar. En caso de que el sistema la encuentre, nos mostrará un mensaje de éxito por pantalla describiendo la labor realizada. En caso contrario, el sistema nos notificará un mensaje de error por pantalla argumentando la razón por la que el borrado no se hubiese podido realizar.

C.7. Hermanos

En dicho panel podemos encontrar funcionalidades asociadas a los hermanos de nuestro sistema dentro Hermandad. Cada funcionalidad será desarrollado en subsecciones de este mismo apartado.

C.7.1. Listado de Hermanos

En esta pantalla podremos encontrar el listado de Hermanos dados de alta en el sistema. En dicha tabla, podemos observar de un primer vistazo datos básicos como DNI, nombre, y apellidos. Se ofrece la posibilidad de exportar la información a Excel y a PDF para poder compartir información con personal externo o realizar métricas/gráficas de la información del sistema. En dicho listado también tenemos la oportunidad de paginar la información por número de registros.

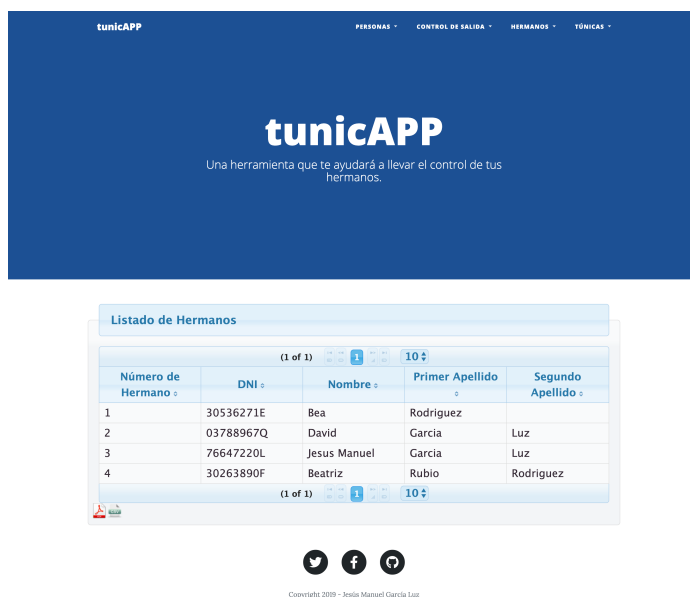


Figura C.9: Pantalla del listado de Hermanos.

C.7.2. Añadir Hermano

En dicha pantalla, tenemos un formulario para introducir un DNI. Dicho DNI debe ser la Persona dada de alta en el sistema previamente en la funcionalidad **Añadir Persona** mencionada con anterioridad.

Recordamos al usuario que un Hermano debe ser una persona del sistema, pero una persona no tiene por qué ser un hermano. Al introducir un DNI, el sistema comprobará si ya existe ese usuario en el sistema. En caso de existir mostrará un mensaje de error describiendo el motivo, y en caso de no existir mostrará los datos no modificables de la persona.

En dicho formulario de alta, podremos decidir el Tipo de Recibo al que se asocia el hermano, el número de cuotas y el Tipo de Pago.

En caso positivo, el sistema habrá dado de alta al usuario, enviando a la dirección de email del mismo una confirmación del alta. En dicho correo, al hermano dado de alta se le facilita un usuario y contraseña para que pueda entrar en el sistema como rol Usuario en el que tendrá acceso a ver **Tus Cuotas** y **Reserva tu Control de Salida**

tunicAPP

Una herramienta que te ayudará a llevar el control de tus hermanos.

Alta de Hermano

Panel de Búsqueda

DNI/NIE (*)
30263891P

Buscar Limpiar

Datos de Alberto Rubio Rodriguez:

DNI/NIE	Número de Documento	
DNI	30263891P	
Nombre	Primer Apellido	
Alberto	Rubio	
Segundo Apellido	Teléfono	Email
Rodriguez	650637377	rubioreatriz@gmail.com
Fecha de Nacimiento		
26/03/1995		
Domicilio		
Esperanza trinidad 5 bloq 11 1c		
Autoriza LOPD		
Si		

Cuota de Hermano

Tipo de Recibo (*)	Número de cuotas (*)
Cuota de Hermano 2019	1
Tipo de Pago (*)	
Metálico	

Dar de Alta

Twitter Facebook GitHub

Copyright 2019 - Jesús Manuel García Luz

Figura C.10: Pantalla para añadir Hermanos.

C.7.3. Editar Hermano

En la pantalla de edición del hermano, tenemos un formulario para introducir el DNI del Hermano. Una vez introducido, el sistema validará el DNI y lo buscará en el sistema dando dos posibles mensajes.

En caso de búsqueda exitosa, el sistema mostrará en pantalla los datos del Hermano. Existen campos editables y campos no editables como son El tipo de documento, el número de documento, la autorización LOPD o el tipo de Recibo entre otros.

En caso de error en la búsqueda, el sistema notificará por pantalla dicho error.

Además, el sistema ofrecerá la posibilidad de editar la Cuota de Hermano. Dicha cuota permite modificar el número de cuotas, el tipo de pago, el estado del pago y la posibilidad de establecer como pagado el recibo.

C.7.4. Eliminar Hermano

En la pantalla de Eliminar Hermano, nos encontraremos un formulario para introducir el DNI del Hermano a borrar. El sistema mostrará dos tipos de mensajes en función al resultado del borrado. En caso de éxito, el hermano dejará de formar parte del sistema. En caso de error, el sistema notificará mediante pantalla el error producido.

C.8. Control de Salida

En dicho panel podemos encontrar funcionalidades asociadas a los controles de salida de nuestro sistema dentro Hermandad. Cada funcionalidad será desarrollado en subsecciones de este mismo apartado.

C.8.1. Listado de Control de Salida

En dicha pantalla tenemos el listado de los controles de salida existentes en la aplicación divididos por titular. Por un lado tenemos los controles de salida asociados al Cristo, y por otro lado tenemos los Controles de salida asociados a la Virgen. Para cada uno de ellos, se ofrece la posibilidad de exportar la información a Excel y a PDF para poder compartir información con personal externo o realizar métricas/gráficas de la información del sistema. En dicho listado también tenemos la oportunidad de paginar la información por número de registros. Además, se ofrece un contador total para saber el sumatorio total de Controles de Salida del año en curso.

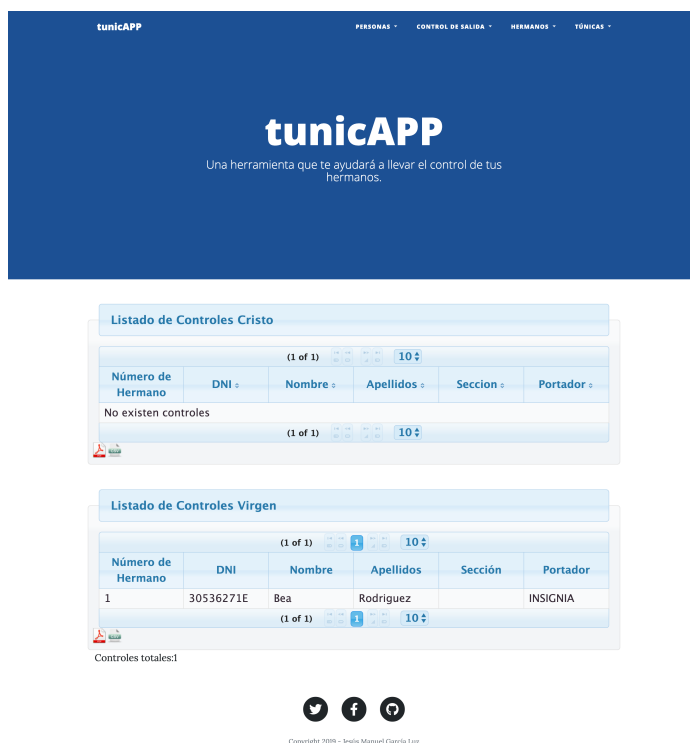


Figura C.11: Pantalla del Listado de Controles de Salida.

C.8.2. Añadir Control de Salida

En la pantalla para añadir un control de salida, el sistema nos ofrece un formulario para buscar un DNI del Hermano al que queremos dar de alta un control de salida. Para dar de alta al hermano, debe tener su cuota del año en curso pagada. Una vez realizada la búsqueda el sistema muestra

dos posibles notificaciones: éxito, donde el sistema muestra los datos del Hermano en cuestión y la posibilidad de añadirle un recibo del tipo Salida Procesional, y error donde el sistema notifica mediante mensaje de pantalla el error producido.

En caso de éxito, veremos los datos del Hermano como hemos mencionado anteriormente. Entre los campos mostrados, tenemos el Recibo de Control de Salida. En dicho formulario, tenemos seleccionar el Tipo de Recibo y automáticamente el sistema aportará la información de cuantía asociada a la Salida Procesional y nos mostrará un formulario sobre la túnica que el hermano desea llevar. Las túnicas se diferencian entre Cristo y Virgen, y sólo saldrán disponibles las túnicas que no posean otros hermanos. Además, se podrá seleccionar la insignia a portar por el hermano.

Cuando realizamos todas las acciones descritas, el sistema hará los cambios en nuestra base de datos y notificará la acción realizada. Dicha acción repercute en un correo al hermano donde se le indica el alta de control de salida, con un justificante adjuntado diseñado por la Hermandad.

The screenshot displays the 'tunicAPP' web interface. At the top, a navigation bar includes links for 'PERSONAS', 'CONTROL DE SALIDA', 'HERMANOS', and 'TÚNICAS'. The main header features the 'tunicAPP' logo and the tagline 'Una herramienta que te ayudará a llevar el control de tus hermanos.' Below this, the title 'Alta Control de Salida' is centered. The form is divided into several sections: 1. 'Panel de Búsqueda' with a 'DNI/NIE (*)' field containing '76647220L' and 'Buscar'/'Limpiar' buttons. 2. 'Datos de Jesus Manuel Garcia Luz:' showing 'DNI/NIE' as '76647220L' and 'Número de Documento' as '76647220L'. 3. 'Control de Salida' with a 'Tipo de Recibo (*)' dropdown set to 'Salida Procesional 2019', a 'Cuantía' field set to '20', and a 'Pagado' dropdown set to 'No'. 4. 'Túnica' section with 'Titular (*)' and 'Portador (*)' dropdowns, both set to 'Seleccione', and a 'Túnica (*)' dropdown set to 'Seleccione'. A 'Dar de Alta el Control' button is at the bottom. Social media icons for Twitter, Facebook, and GitHub are present, along with a copyright notice: 'Copyright 2019 - Jesús Manuel García Luz'.

Figura C.12: Pantalla para añadir un Control de Salida.

C.8.2.1. Justificante Control de Salida

El justificante generado por la aplicación y enviado por email es un diseño estático que, en acuerdo con el cliente, se puede ir modificando año tras año con el versionado de la aplicación.



Figura C.13: Justificante generado al Control de Salida.

C.8.3. Editar Control de Salida

Para la edición del control de salida, el sistema nos ofrece la oportunidad de introducir el DNI del Hermano al que queremos editar el control de salida. Dicha acción genera dos tipos de notificaciones en el sistema. La exitosa, hará que el sistema muestre los datos asociados a dicho DNI pudiendo cumplimentar y establecer como pagado el control de salida resultante. La notificación errónea mostrará por pantalla un mensaje con el error asociado. Además, en esta funcionalidad podemos añadir a la sección que irá dicho Hermano. Cualquier modificación en el control de salida generará un email que será enviado al Hermano por si su control de salida se viese actualizado.

The screenshot displays the tunicAPP web interface. At the top, a dark blue header contains the logo 'tunicAPP' and the tagline 'Una herramienta que te ayudará a llevar el control de tus hermanos.' Below the header, a navigation bar lists 'PERSONAS', 'CONTROL DE SALIDA', 'HERMANOS', and 'TÚNICAS'. The main content area is titled 'Editar Control de Salida' and contains several form sections:

- Panel de Búsqueda:** Includes a 'DNI/NIE (*)' field with the value '30536271E' and 'Buscar' and 'Limpiar' buttons.
- Datos de Bea Rodriguez :** A table with two columns: 'DNI/NIE' and 'Número de Documento'. The 'DNI' field contains '30536271E'. Below the table are fields for 'Nombre' (Bea), 'Apellidos' (Rodriguez), and a 'Rodriguez' field.
- Control de Salida:** Includes a 'Tipo de Recibo (*)' dropdown set to 'Salida Procesional 2019', a 'Cuantía' field with '20', a 'Pagado' dropdown set to 'No', and an 'Establecer como pagado' button.
- Túnica:** Includes a 'Titular' dropdown set to 'VIRGEN', a 'Túnica' dropdown set to 'V-3 de 123cm por 123cm', a 'Portador' dropdown set to 'Seleccione', and a 'Sección' field.

At the bottom of the form is an 'Editar Control de Salida' button. Below the form are social media icons for Twitter, Facebook, and YouTube, and a copyright notice: 'Copyright 2019 - Jesús Manuel García Luz'.

Figura C.14: Editar un Control de Salida

C.8.4. Eliminar Control de Salida

Para la eliminación de un Control de Salida, el usuario deberá introducir el DNI del Hermano al que quiere eliminar el control de salida. El sistema generará dos tipos de notificaciones para esta acción: mensaje mostrado en pantalla positivo para el éxito en el borrado, y mensaje mostrado en pantalla negativo para el posible error.

C.8.5. Estadísticas

Para la obtención de información de forma rápida y coherente, la aplicación incorpora un módulo de estadísticas. Si queremos representar información, jamás estaremos conformes de la cantidad de información representada... y por ello, se ha decidido realizar una muestra de cómo y qué tipo de información podemos representar que represente una información útil al usuario.

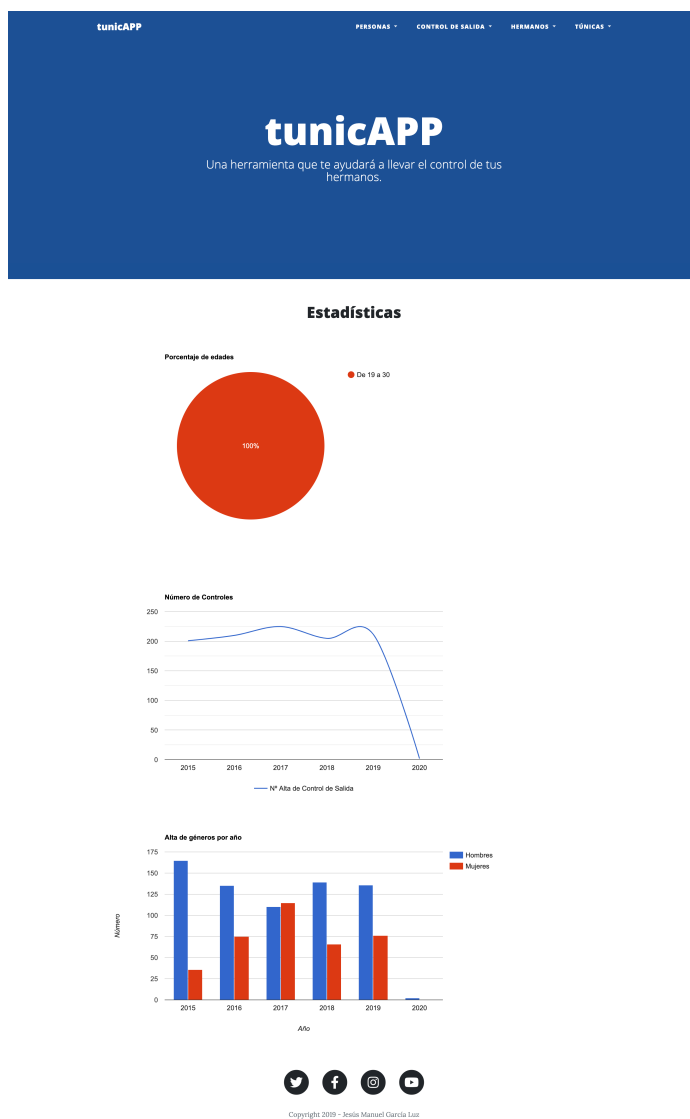


Figura C.15: Estadística

C.9. Pantalla principal Usuario

Una vez introducido el usuario y la contraseña en la pantalla de Login, obtenemos la pantalla de Usuario en el caso de que el rol introducido fuese de administración. En dicha pantalla podemos visualizar información sobre la aplicación y sus diferentes opciones situadas en el panel superior de la pantalla. Dependiendo de la funcionalidad que queramos trabajar tenemos representado:

- Tus Cuotas
- Reserva tu Control de Salida

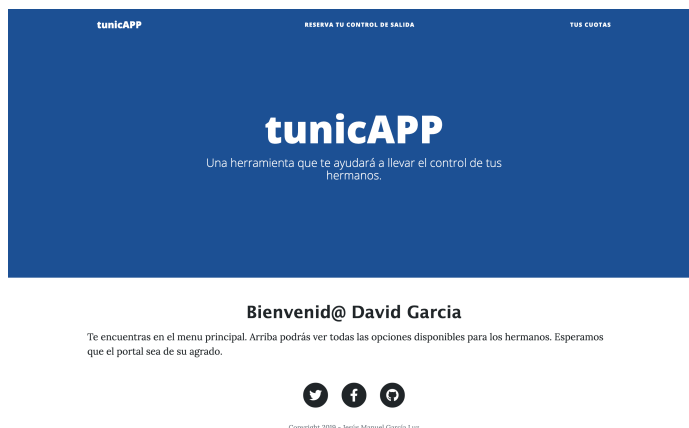


Figura C.16: Pantalla de Inicio para rol Usuario.

C.9.1. Tus Cuotas

En este apartado, el hermano podrá visualizar su ficha de Hermano. En dicha ficha, todos los campos son no editables evitando así que el hermano pueda modificar sus datos sin el consentimiento o apoyo del gestor de la aplicación.

Lo único que se puede editar es el Tipo de Pago. Si dejamos el tipo de pago Metálico, será el gestor de la Hermandad el que marcará como pagado en su ficha el control cuando el hermano acuda a la casa de Hermandad a pagarlo.

En caso de optar por un pago mediante Transferencia Bancaria, el hermano rellenará la cuenta bancaria y guardará sus cambios. Esto genera una notificación para que la Hermandad, pueda incluir en su nómina de domiciliaciones dicha cuenta bancaria y en el caso de ser recibido su cobro, poder editar la ficha de Hermano y marcarla como pagado.

Por otro lado, si el hermano selecciona la opción Paypal, el sistema se comunicará con el paypal del Hermano y en el caso de que PayPal nos de el visto bueno de la transacción, se marcará como Pagado el recibo.

Todas estas acciones se verán reflejadas en un correo a la Hermandad al hacer click en Editar.

C.9.2. Reserva tu Control de Salida

En dicha pantalla, se le ofrece al Hermano poder realizar la reserva del Control de Salida desde el portal. Para ello, el sistema le muestra un formulario a cumplimentar.

En dicho formulario aparecen sus datos básicos no editables y en Control de Salida a cumplimentar. En cuanto seleccione el control de Salida, el sistema autocompletará los campos asociados al mismo de forma automática como la cuantía.

tunicAPPRESERVA TU CONTROL DE SALIDATUS CUOTAS

tunicAPP

Una herramienta que te ayudará a llevar el control de tus hermanos.

Tu ficha de Hermano

Datos de David Garcia Luz:

DNI/NIE

DNI

Número de Documento

03788967Q

Nombre

David

Primer Apellido

García

Segundo Apellido

Luz

Teléfono

666666666

Email

yisekai@gmail.com

Fecha de Nacimiento

06/10/1994

Domicilio

Calle Granada 4, Piso 4 Puerta 2

Autoriza LOPD

Si

Cuota de Hermano

Tipo de Recibo

Cuota de Hermano 2019

Número de cuotas

1

Pagado

Si

Editar

Copyright 2019 - Jesús Manuel García Luz

Figura C.17: Pantalla de Tus Cuotas.

Además, el sistema le ofrece las tres posibilidades de pago y la posibilidad de comprobar su pago ya que, con los pagos de transferencia bancaria o metálico, la actualización del pago depende de terceros y es necesario consultar el estado del pago.

Posteriormente el Hermano podrá seleccionar Cristo o Virgen y las túnicas disponibles para el mismo. Tras ello, le dará click a Dar Alta de Control y se registrará la información notificando mediante email la operación.

Bibliografía

- [1] QUÉ ES JSF, <http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion01-apuntes.html> Titulo de Experto Universitario en Desarrollo de Aplicaciones y Servicios con Java EE por la Universidad de Alicante.
- [2] QUÉ ES JPA, <https://www.arquitecturajava.com/ejemplo-de-jpa/> Arquitectura Java
- [3] QUÉ ES PRIMEFACES, <https://www.primefaces.org>
- [4] REQUISITOS NO FUNCIONALES, https://www.ecured.cu/Requisitos_no_funcionales
- [5] CÓMO INTEGRAR EL API PAYPAL, <https://developer.paypal.com/docs/>
- [6] CÁLCULO DEL DÍGITO DE CONTROL DEL NIF/NIE, <http://www.interior.gob.es/web/servicios-al-ciudadano/dni/calculo-del-digito-de-control-del-nif-nie>
- [7] CURSO AVANZADO DE PROGRAMACIÓN EN JAVA EE: STRUTS, JSF, AJAX, EJB, JPA (ESPAÑOL)
Ramón Egido García con ISBN-10: 8461606612
- [8] SELENIUM DEV <https://www.selenium.dev>
- [9] DUDAS GENERALES SOBRE EL DESARROLLO, <https://stackoverflow.com>,
- [10] ORACLE SQL, <https://www.oracle.com/es/index.HTML>,
- [11] L^AT_EX PARA DOCUMENTACIÓN, <https://en.wikibooks.org/wiki/LaTeX/>,
- [12] GOOGLE CHARTS, <https://developers-dot-devsite-v2-prod.appspot.com/chart/interactive/docs>